

8-16-2017

FATREC Workshop on Responsible Recommendation Proceedings

Michael Ekstrand
Boise State University

Amit Sharma
Microsoft



FATREC Workshop on Responsible Recommendation

TABLE OF CONTENTS

PROCEEDINGS OVERVIEW	2
INTRODUCTION	2
WORKSHOP PROGRAM.....	2
COMMITTEES	3
TOWARDS MINIMAL NECESSARY DATA	4
ON QUANTIFYING KNOWLEDGE SEGREGATION IN SOCIETY.....	10
BALANCED NEIGHBORHOODS FOR	
FAIRNESS-AWARE COLLABORATIVE RECOMMENDATION	14
IMPACT OF TASK RECOMMENDATION SYSTEMS IN CROWDSOURCING	
PLATFORMS.....	19
THE PRICE OF FAIRNESS IN LOCATION BASED ADVERTISING	25
FAIR SHARING FOR SHARING ECONOMY PLATFORMS.....	30
EXPLORING EXPLANATIONS FOR MATRIX FACTORIZATION	
RECOMMENDER SYSTEMS	34
DO NEWS CONSUMERS WANT EXPLANATIONS FOR PERSONALIZED NEWS	
RANKINGS?	38
PRESENTING DIVERSITY AWARE RECOMMENDATIONS	44
ACADEMIC PERFORMANCE PREDICTION	
IN A GENDER-IMBALANCED ENVIRONMENT	48
CONSIDERATIONS ON RECOMMENDATION INDEPENDENCE	
FOR A FIND-GOOD-ITEMS TASK.....	52

FATREC Workshop on Responsible Recommendation

Proceedings Overview

INTRODUCTION

We sought, with this workshop, to foster a discussion of various topics that fall under the general umbrella of *responsible recommendation*: ethical considerations in recommendation, bias and discrimination in recommender systems, transparency and accountability, social impact of recommenders, user privacy, and other related concerns. Our goal was to encourage the community to think about how we build and study recommender systems in a socially-responsible manner.

Recommendation systems are increasingly impacting people's decisions in different walks of life including commerce, employment, dating, health, education and governance. As the impact and scope of recommendations increase, developing systems that tackle issues of fairness, transparency and accountability becomes important. This workshop was held in the spirit of FATML (Fairness, Accountability, and Transparency in Machine Learning), DAT (Data and Algorithmic Transparency), and similar workshops in related communities. With "Responsible Recommendation", we brought that conversation to RecSys.

WORKSHOP PROGRAM

Setting the Stage

1. *Towards Minimal Necessary Data: The Case for Analyzing Training Data Requirements of Recommender Algorithms*. Martha Larson, Alessandro Zito, Babak Loni, and Paolo Cremonesi. DOI: [10.18122/B2VX12](https://doi.org/10.18122/B2VX12)
2. *Quantifying Knowledge Segregation in Society*. Abhijnan Chakraborty, Muhammad Ali, Saptarshi Ghosh, Niloy Ganguly, and Krishna P. Gummadi. DOI: [10.18122/B2SK5H](https://doi.org/10.18122/B2SK5H)
3. *Balanced Neighborhoods for Fairness-Aware Collaborative Recommendation*. Robin Burke, Nasim Sonboli, Masoud Mansoury, and Aldo Ordonez-Gauger. DOI: [10.18122/B2GQ53](https://doi.org/10.18122/B2GQ53)

Measuring Fairness and Tradeoffs

1. *Impact of Task Recommendation Systems in Crowdsourcing Platforms*. Kathrin Borchert, Matthias Hirth, Steffen Schnitzer, and Christoph Rensing. DOI: [10.18122/B2CX1Q](https://doi.org/10.18122/B2CX1Q)
2. *Price of fairness for location based advertising*. Christopher Riederer and Augustin Chaintreau. DOI: [10.18122/B2MD8C](https://doi.org/10.18122/B2MD8C)
3. *Fair Sharing for Sharing Economy Platforms*. Abhijnan Chakraborty, Joanna Asia Biega, Aniko Hannak, and Krishna P. Gummadi. DOI: [10.18122/B2BX2S](https://doi.org/10.18122/B2BX2S)

Explanations and Persuasion

1. *Explanation Strategies for Matrix Factorization Recommender Systems*. Bashir Rastegarpanah, Mark Crovella, and Krishna Gummadi. DOI: [10.18122/B2R717](https://doi.org/10.18122/B2R717)
2. *Do News Consumers Want Explanations for Personalized News Rankings?*. Maartje Ter Hoeve, Mathieu Heruer, Daan Odijk, Anne Schuth, Martijn Spitters, Ron Mulder, Nick van der Wildt, and Maarten de Rijke. DOI: [10.18122/B24D7N](https://doi.org/10.18122/B24D7N)
3. *Presenting Challenging Recommendations: Making Diverse News Acceptable*. Nava Tintarev. DOI: [10.18122/B2HQ41](https://doi.org/10.18122/B2HQ41)

Fair Algorithms

1. *Academic performance prediction in a gender-imbalanced environment*. Piotr Sapiezynski, Valentin Kassarnig, Christo Wilson, Sune Lehmann, and Alan Mislove. DOI:[10.18122/B20Q5R](https://doi.org/10.18122/B20Q5R)
2. *Consideration on Recommendation Independence for a Find-Good-Items Task*. Toshihiro Kamishima and Shotaro Akaho. DOI:[10.18122/B2871W](https://doi.org/10.18122/B2871W)

ORGANIZING COMMITTEE

- Michael Ekstrand (Boise State University)
- Amit Sharma (Microsoft Research)

PROGRAM COMMITTEE

- Solon Barocas (Microsoft Research, Cornell University)
- Pablo Castells (Universidad Autónoma de Madrid)
- Ed H. Chi (Google)
- Dan Cosley (Cornell University)
- Jennifer Golbeck (University of Maryland)
- Krishna Gummadi (MPI-SWS)
- Daniel Kluver (University of Minnesota / Macalester College)
- Bart Knijnenburg (Clemson University)
- Martha Larson (Radboud University / Delft University of Technology)
- Maria Soledad Pera (Boise State University)
- Pierre-Nicolas Schwab (IntoTheMinds / RTBF)
- Suresh Venkatasubramanian (University of Utah)

Towards Minimal Necessary Data: The Case for Analyzing Training Data Requirements of Recommender Algorithms

Martha Larson^{1,2}, Alessandro Zito³, Babak Loni², Paolo Cremonesi³

¹Radboud University, Netherlands

²Delft University of Technology, Netherlands

³Politecnico di Milano, Italy

m.larson@cs.ru.nl, zito.ales@gmail.com, b.loni@tudelft.nl, paolo.cremonesi@polimi.it

ABSTRACT

This paper states the case for the principle of minimal necessary data: If two recommender algorithms achieve the same effectiveness, the better algorithm is the one that requires less user data. Applying this principle involves carrying out training data requirements analysis, which we argue should be adopted as best practice for the development and evaluation of recommender algorithms. We take the position that responsible recommendation is recommendation that serves the people whose data it uses. To minimize the imposition on users' privacy, it is important that a recommender system does not collect or store more user information than it absolutely needs. Further, algorithms using minimal necessary data reduce training time and address the cold start problem. To illustrate the trade-off between training data volume and accuracy, we carry out a set of classic recommender system experiments. We conclude that consistently applying training data requirements analysis would represent a relatively small change in researchers' current practices, but a large step towards more responsible recommender systems.

ACM Reference format:

Martha Larson^{1,2}, Alessandro Zito³, Babak Loni², Paolo Cremonesi³. 2017. Towards Minimal Necessary Data: The Case for Analyzing Training Data Requirements of Recommender Algorithms. In *Proceedings of Workshop on Responsible Recommendation at ACM RecSys'17, Como, Italy, 31 August 2017 (FATREC'17)*, 6 pages.
<https://doi.org/10.18122/B2VX12>

1 INTRODUCTION

Conventionally, recommender algorithms are developed to exploit all available training data. Although there is wide-spread awareness of the downsides of such *data greed* during algorithm training and deployment, the convention stands largely unquestioned. In other words, researchers generally know that prediction performance saturates after a certain amount of data has been collected from users, and additional data only increases training times. However, this knowledge is currently not translated into best practice for the development of recommender systems algorithms.

In this paper, we state the case for the practice of analyzing training data requirements during the development and evaluation of recommender system algorithms. Such an analysis implements the principle of minimal necessary data: If two recommender algorithms achieve the same effectiveness, the better algorithm is

the one that requires less user data. Our position is that any new recommender algorithm should be judged by the way in which it trades off between accuracy and amount of training data used. Beyond a certain point, additional training data will not have a meaningful effect on predictions. Effectively, the extra training data will have an “invisible” impact on user experience. We argue that pushing the collection and use of user data beyond this point should be discouraged. In short, a responsible recommender system takes no more from users than it needs to. The case for training data requirements analysis is closely related to the 2013 idea of *Differential Data Analysis* [6], which creates characterizations of which data contributes most to the accuracy of a recommender algorithm. The extended arXiv version of [6] emphasizes that data is a liability: services providers need to protect it, and they need to respond to subpoenas. Data breaches are a serious worry for companies storing data. Considerations of privacy and data security are becoming increasingly important as Europe continues to emphasize users controlling their own personal data (cf. the EU General Data Protection Regulation¹, which goes into force in 2018).

With this paper, we build on the motivation of [6], and also echo the question, “Is all this data really necessary for making good recommendations?” We first argue for the importance of training data requirements analysis in recommender system research. Then, we report on classic experiments showing that lengthening the history-length of the training set does not necessarily improve prediction accuracy. The picture that emerges is that recommender systems have much to gain, and actually nothing to lose, in moving towards minimal necessary data.

2 BACKGROUND AND MOTIVATION

This section looks at aspects of the current state of recommender system research that motivate minimal necessary data.

Addressing the Data Greed Habit Looking at the field of recommender system research and development as a whole, unquestioned data greed is quite surprising. We point to the work on cold-start recommendation, and in particular to [8], as evidence that researchers are well aware that after a certain saturation point more data does not necessarily translate into better performance. We suspect that data greed is simply a bad habit developed when standard, static data sets are used for evaluation. With such data sets the assumption that “more is always better” does not lead to any obvious negative consequences. On the contrary, comparison of results on standard data sets requires standardized test/training

This article may be copied, reproduced, and shared under the terms of the Creative Commons Attribution-ShareAlike license (CC BY-SA 4.0).

FATREC'17, 31 August 2017, Como, Italy

© 2017 Copyright held by the owner/author(s).

DOI: 10.18122/B2VX12

¹<http://ec.europa.eu/justice/data-protection>

splits. In other words, using less than all available data is actually associated with faulty methodology. Adopting training data analysis as a best practice would maintain comparability between research results, while at the same time allowing application of minimal necessary data.

Fulfilling Non-Functional Requirements Recent years have seen a push towards evaluating recommender systems with respect to not only functional, but also non-functional requirements [26]. During this time, analysis of resource use has become more common in the literature, and the development of algorithms with unnecessary computational complexity or high response time has been discouraged. Training data requirements analysis is another form of resource analysis that supports understanding of the practical usefulness of recommender algorithms in real-world settings. Seen in this way, minimal necessary data is a continuation of an existing evolution.

Ensuring User-centered Recommendation Recently, research studies have demonstrated that algorithm accuracy does not necessarily play a dominant role in the reception of a recommender system by users [7, 10]. If performance improvements achieved by using more data to train a recommender system are too slight or subtle for users to notice, the additional data is adding no value, and should not be used. We understand responsible recommendation as recommendation that serves the people whose data it uses. Conscientious service of users requires formulating an explicit definition of success that characterizes the goals of the recommender system. The definition should contain a specification of the trade-off between accuracy levels and user experience. Such a definition throws a spotlight on where recommender systems are collecting, storing, and using data that is not needed. Using more data than needed imposes on users' privacy, and, collecting user data that does not serve a specific goal cannot be justified.

In sum, if the convention of data greed has no principled justification, and the recommender system community is already focusing on non-functional requirements and user experience, it is an obvious and relatively small step to focus on minimal necessary data.

3 RELATED WORK

Here, we overview previous work related to trade-offs between training data volume and recommender system prediction performance.

3.1 Analyzing Training Data Requirements

Papers analyzing training data requirements are scattered throughout the recommender system literature. In 2008, [27] evaluated the performance of algorithms on the Netflix Prize dataset against the number of users in the training data. There is a clear saturation between 100,000-480,000 users, i.e., the algorithm does not achieve continued improvement. The plot is on log scale, and the authors are focused on what can be achieved by 0-100,000, and do not mention the saturation effect. Also in 2008, [21] analyzed the impact of the number of using ratings on news item recommendation. In 2010, [22] analyzed the number of weeks of training data on the recommendations of seminar events at a university. On the whole, we find that attention to minimal necessary data has been the exception rather than the rule.

3.2 Doing More with Less

In addition to work that looks at the impact of training data volume on specific algorithms, other work is dedicated to actually developing algorithms that do more with less. In the general machine learning literature, there is clear awareness that certain algorithms are better suited than others for performing under conditions of limited data, e.g., [11]. Here, we mention some other examples of work that we are closely connected to. Cold start is the classic case in which recommender system algorithms must be capable of doing more with less. Different sizes of datasets have been studied in order to investigate different levels of cold start [8, 9]. Further [8] shows that there is a difference between algorithms with respect to data requirements. The idea of minimal necessary data can be seen as the proposal to take the ability of algorithms designed to address cold start conditions and applying it as broadly as possible.

In [23], we touched on the privacy benefits of algorithms that do not need to store data in association with user IDs for long periods. Explicit attention to minimal necessary data will promote the development of such algorithms. We note that algorithms that use minimal personal data are useful to address news recommendation, where user IDs might be unstable or unavailable [16].

3.3 Timed-based Training Data Analysis

The closest work to the experiments presented in this paper is work on time-aware recommender systems. The survey article [4] discusses techniques that weight ratings by freshness and mentions that the more extreme version of such an approach is *time truncation*, i.e., actually dropping ratings older a specified threshold. They authors cite only two time-truncation papers. The first is [5], which demonstrates that using information near the recommendation date improves accuracy on the CAMRa 2010 Challenge. The second is [13], which reports interesting results using a time-window filtering technique intended to capture fluctuations in seasonal demand for items. Perhaps the most well-known work on time-aware recommendation is Collaborative Filtering with temporal dynamics [18]. Here, we adopt [18] as a baseline to demonstrate the effect of time truncation above and beyond time-based weighting.

4 EXPERIMENTAL SETUP

Next, we turn to a set of experiments that illustrate the trade-off between data volume and prediction accuracy using a timed-based training data requirements analysis. In this section, we describe our data sets, recommender algorithms, and analysis methodology. Our experiments support the position that this trade-off should not be considered a a tweak to be taken care of by engineers at deployment time. Rather, training data size has substantial measurable impact in common experimental set ups used by recommender system researchers. Here, we study time truncation since it is a well-established method for identifying training data that is less valuable. We emphasize that other approaches, such as sampling, are important for training data requirements analysis.

4.1 Data sets

We choose to experiment on three data sets. The data sets were chosen because of their long temporal duration, and the fact that they are widely used, which supports reproducibility. The first two,

MovieLens 10M and NetFlix were selected because they are ‘classic’, in the sense that they are well understood by the community. The third, a dataset from Amazon, is representative of a highly sparse recommender problem.

Basic statistics of the datasets are shown in Table 1. We briefly mention the temporal ranges and further details about each. The MovieLens 10M dataset [14] has a data range from January 1995 to January 2009 (14 years). The Netflix dataset [3] was collected between October 1998 and December 2005 (7 years). To make the dataset size manageable we randomly selected 10% of users. We observed that our sample is big enough to cover almost all the movies and that the distribution of ratings has the same shape in the sample and in the original dataset. Furthermore, the temporal window of the sample is almost as long as the original dataset.

The Amazon dataset [19] consists of ratings collected from June 1995 to July 2005 (10 years). We use only the products that belong to the four main product groups: books, DVDs, music and videos.

Dataset	#Users	#Items	#Ratings	Density
ML-10M	69.878	10.681	10M	4,47%
Netflix	480.180	17.770	100,5M	1,2%
Amazon	1.553.447	401.961	7,5M	0,001%

Table 1: Datasets statistics

4.2 Recommender framework

We use four different algorithms to train our models. The experiments are implemented using WrapRec [20], an open source evaluation framework for recommender systems. The experiment were run on a machine with 16 CPU cores with clock speed of 2.3 GHz and 16 GB of memory. The following algorithms are used in this work where the first three are used for rating prediction and the last one is used for the top-N ranking task.

Biased Matrix Factorization (BMF): This method [18] is the most widely-used model-based algorithm for rating prediction problems. This method is the standard Matrix Factorization model with user, item and global biases. In this work, we used the MyMediaLite [12] implementation of BMF with its default hyper-parameter values. The optimization algorithm is Stochastic Gradient Descent (SGD) with a learning rate of 0.01. The latent factors are initialized with a zero-mean normal distribution with standard deviation of 0.1. The number of latent factors, however, is varied. Our experiments demonstrate the effect of latent factors.

Factorization Machines: Factorization Machines [24] are state-of-the-art models for rating prediction problems. In this work, we used the more advanced optimization method of Markov Chain Monte Carlo (MCMC), that is implemented in LibFm [24]. The only hyper-parameter of the MCMC algorithm, i.e., the standard deviation of the initializer distribution, is set to 0.1, the default value in the LibFm implementation [24].

Time-Aware Factor Model: This method [17] is also a latent factor model for rating prediction problems. The temporal effect of user preferences is modeled with a time-dependent bias function. This method yielded top performance in the Netflix prize. The hyper-parameters are the default values of the MyMediaLite implementation of Time-Aware model.

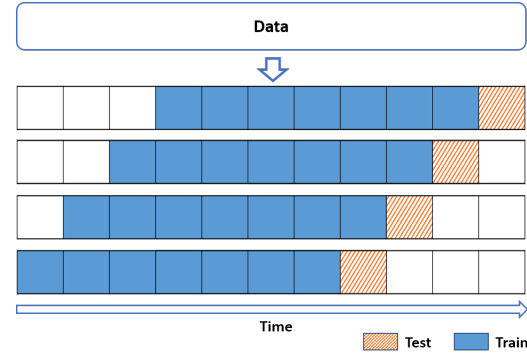


Figure 1: An Overview of the sliding process.

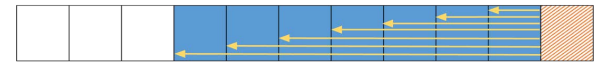


Figure 2: Representation of sliding window for one fold.

Bayesian Personalized Ranking (BPR): This method [25] is an state-of-the-art method for ranking problems where the learning involves optimization for ranking. Since this method is designed for datasets with unary positive-only feedback, we consider the ratings above user average rating as a positive feedback. BPR uses SGD for optimization. The learning rate is set to 0.05 and the standard deviation of the initializer is set to 0.1.

4.3 Sliding-window Cross-validation

Our experiments use *sliding window cross-validation*, which allows us to maintain the temporal ordering of the data (also referred to as ‘forward chaining’). We start by partitioning the data into 11 temporal segments. Each fold of the cross-validation consists of a data window that is split into test and training data. The test data consists of the temporally most recent segment. To create multiple folds, the data window is slid backwards in time by one segment, such that the test data is different for each fold. The sliding process is illustrated in Figure 1. We vary the size of the training dataset by increasing its *history length*, i.e., the length of time that the training dataset extends into the past. We test seven history lengths, indicated by the arrows in Figure 2. Each history length is created by adding one segment to the next-shortest history length. Since our initial split created 11 segments, increasing the history length by one segment means increasing the training data by 10%.

Our data partitioning method makes it possible to validate the results using a training set up to the length of seven segments preceding the test set in each fold. We could have extended the training set with additional segments, but, as we will see in the next section, seven are sufficient to illustrate the phenomenon of saturation that motivates our research. We also noted that the different datasets have different trends in density development as the history length of the training set grows longer. Although we do not measure it formally here, this gives us confidence that the effects we observe are not caused by density trends.

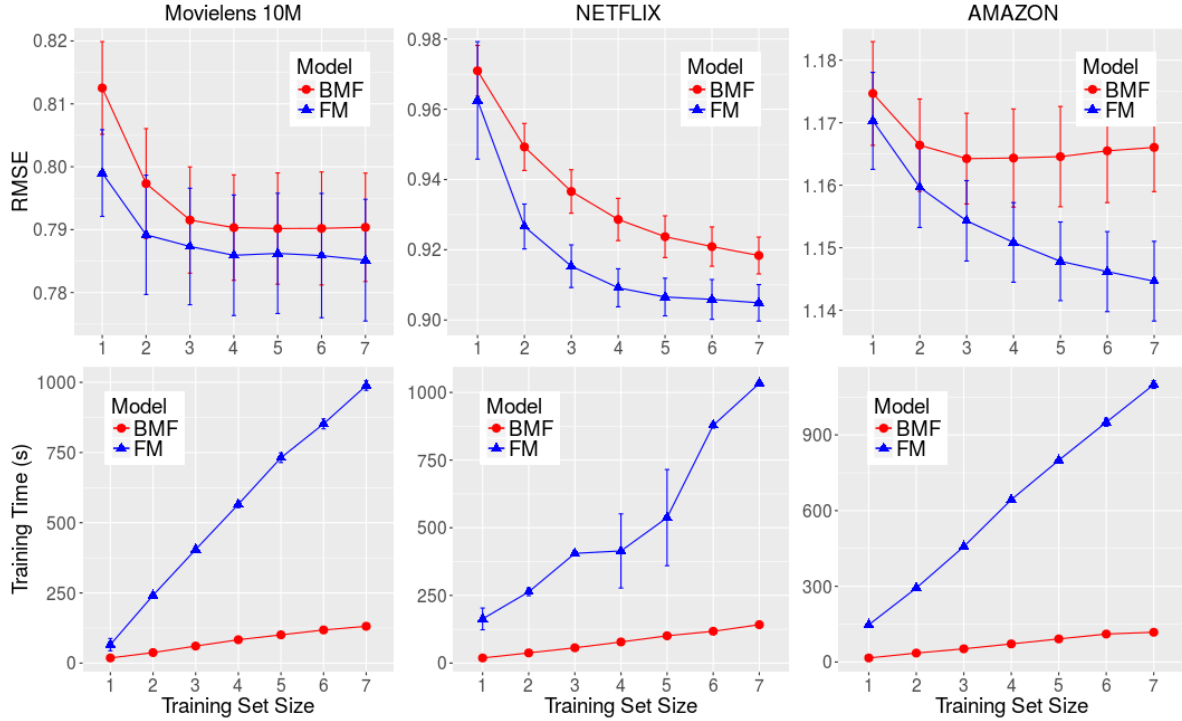


Figure 3: Empirical comparison of the performance and the training time of the two methods of BMF and FM on our three datasets with respect to the training set size (history-length of training set measured in segments).

5 EXPERIMENTS AND RESULTS

We perform three different experiments in order to observe the effect of increasing the history length based on different models.

5.1 Impact of Training Data History-Length

In this experiment, we apply our recommender algorithms while increasing the training set size by extending the history-length of the training window (see Figure 1). The experiment has a relatively a naïve formulation: we simply observe what happens when we apply time-truncation when training classic recommender system algorithms out of the box. For this experiment, we used Biased Matrix Factorization (BMF) and Factorization Machines (FM). The performance of the models are evaluated using the Root Mean Squared Error (RMSE) metric. We also measured the training time of the two models based on different number of segments. Both models were trained with 30 iterations and 10 latent factors. As can be seen in Figure 3, as the history length of the training dataset increases, a certain saturation effect can be observed with all three datasets. At the same time, the training time increases linearly with the history length. The saturation is quite dramatic with MovieLens 10M. However, in all cases there is a clear fall off in the added value of extra data once the training set reaches a certain size. These results show that a large reduction of training data requires a relatively small trade-off of prediction accuracy.

Next, we dive more deeply to investigate whether the choice of the number of latent factors explains the saturation effect. The left column of Figure 4 shows the influence of the number of latent

factors. For this experiment, we use the BMF model and two data sets, MovieLens 10M and Netflix. The figures confirm that the saturation effect dominates the impact of the choice in the number of factors. In other words, increasing the number of latent factors does not necessarily cause the model to benefit from more data.

5.2 Exploiting Temporal Dynamics

In this section, we look more closely at temporal effects. The purpose of this experiment is to eliminate the possibility that the observations in the previous section can be attributed to time-truncation acting as a primitive method for incorporating temporal dynamics into a model. We use the time-aware factor model, introduced in [17], where the temporal aspect of user preferences are exploited using a time-dependent bias function. We use same procedure as in previous experiments to increase the size of the training set. The middle column of Figure 4 reports results on the MovieLens 10M and Netflix datasets. The fact that we find saturation effects using an algorithm that models temporal dynamics, suggests that time-truncation of training data should be used in addition to exploiting temporal dynamics.

5.3 Top-N Recommendations

Next, we turn to Top-N Recommendation and explore the effect of training set size on a learning-to-rank method. We used Bayesian Personalized Ranking (BPR) [25] to train our model, and report results in terms of recall at three different cut-off levels N . We used same number of iterations and latent factors as the naïve experiment (Section 5.1). To calculate recall, we apply a procedure

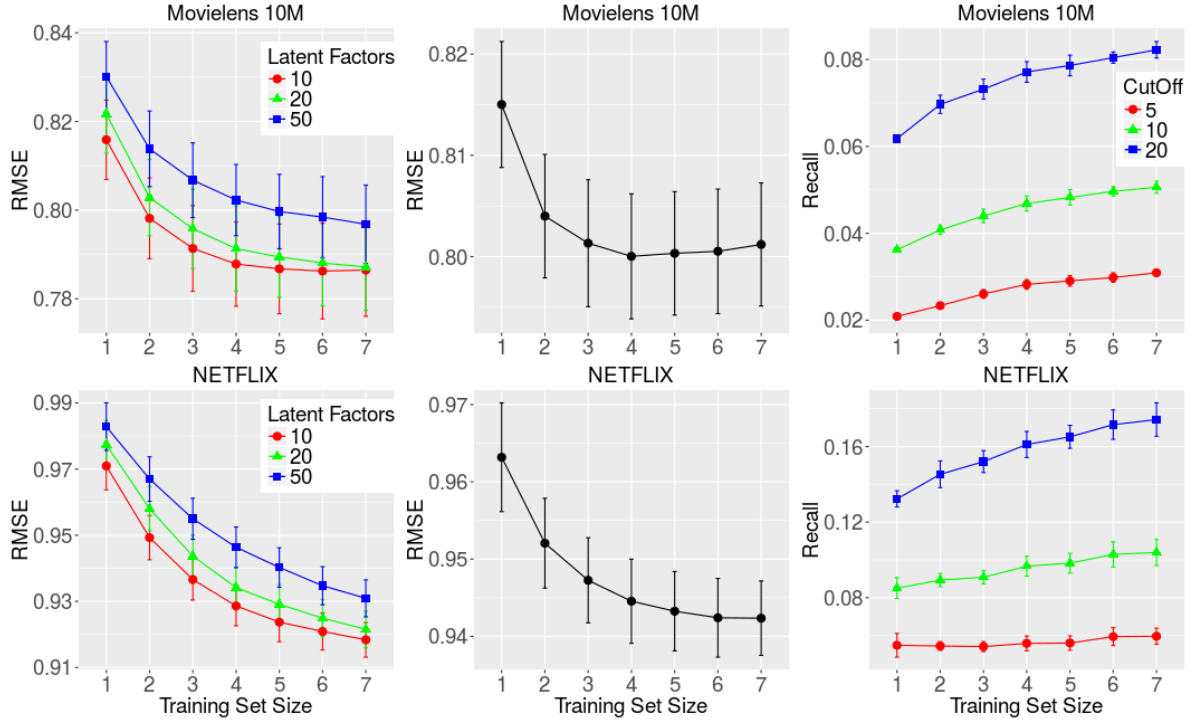


Figure 4: The effect of number of latent factors (left column), saturation effect on the time-aware latent factor model (middle column) and the effect of history-length size on a ranking model (right column)

known as *one-plus-random* [9]: For each test point, 1000 random items that are not rated by the user are sampled and the target item is added. This set of 1001 items are ranked using the trained model. If the target item appears in one of the top-N positions of the ranked list, we count that test point as having achieved a hit. As can be seen in the right column of Figure 4, a smaller training set benefits more from additional data than a larger training set. The results are in this way comparable to what we found in our rating prediction experiments in Sections 5.1 and 5.2. We also calculated performance with respect to Mean Reciprocal Rank (MRR), which is not depicted here for space reasons. For both recall and MRR, we observe diminishing returns effects.

5.4 Discussion

Our experiments illustrate saturation effects as the size of the training dataset increases, but also reveal aspects of data reduction that are not yet thoroughly understood. Following the idea of Differential Data Analysis [6], we would like to have insight into when and why we observe saturation, i.e., diminished returns from additional data. Users in the test set are likely to be represented in the segments temporarily closest to the test set. Ideally, we would like to understand how much of the effect can be attributed to pruning inactive users, and how much is related to taste/item shift, or its opposite, information redundancy. A detailed understanding of these effects would make it possible to design schemes for data collection and retention that have minimal impact on user privacy. For example, if inactive users are no longer contributing to improving predictions, their data should simply be deleted.

When to apply time-truncation is not easy to predict. During our exploratory experiments, we found that prediction accuracy using the smaller data set MovieLens 1M, with 1M ratings and a time span of 3 years (leading to much shorter segments than with ML 10M), does not saturate. This effect suggests that further investigation is needed into the relationship between training data history length and performance for shorter history lengths. We believe, however, that very recent history is very valuable. For example, [15] demonstrates the value of adding information on the most recent history items that the user has interacted with to the prediction for the current item using ML 1M. To better support privacy, we would like to give further consideration to user-specific data dropping, i.e., truncating specific user histories when certain conditions hold. For example, future research could focus on optimizing algorithms that exploit only the very most recent interactions of the user, and delete older interactions. Our initial experiments in this area revealed that it is not trivial. User truncation, could, however, ultimately lead to recommender systems that are not only privacy-sensitive, but also more even handed, and do not favor active users.

6 CONCLUSION AND OUTLOOK

In this paper, we have made a case for recommender systems research to adopt training data requirements analysis as a best practice when developing and evaluating new algorithms. Specifically, researchers developing a recommender system should explicitly analyze the trade-off between the amount of data that the system requires, and the performance of the system. When the improvement in prediction performance becomes negligible, more data

should not be used. If two algorithms achieve the same prediction performance, the algorithm that uses less data should be preferred.

We have presented experimental evidence that trade-offs between objective metrics and the amount of data used deserve increased attention in recommender research. We argue that the recommender system community is well aware of results of this sort, and implicitly already understands the disadvantages of *data greed* and also of the benefits of doing more with less. Carrying out an analysis that demonstrates that an algorithm uses minimal necessary data represents a straightforward application of this awareness. A relatively small shift in research practices represents a large step towards more responsible recommender systems.

As mentioned in the introduction, there is a connection between algorithms that determine the usefulness of data, and user privacy [6]. Obfuscation can protect users and does not necessarily impact recommender performance. Techniques involving obfuscation have been used to anonymize data sets, enabling their release for research purposes, as in [2]. Moving forward, we feel that the idea of minimal necessary data can provide an entry point for researchers in becoming interested in developing obfuscation techniques.

We close with a warning about adopting the position that ‘someone else is doing it’. A metareviewer of a previous version of this paper commented, “How to obtain good recommendations from a minimal amount of data is an interesting problem. At the same time, the idea the the predictive modeling performance improves as the training data grows but eventually tends to level off has been well established in machine learning and is quite well understood (i.e., the concept of learning curves in machine learning reflects exactly that).” We agree with this statement. A recent article in *The Economist* [1] quotes Google’s chief economist commenting on the “decreasing returns to scale” of data. However, we are left wondering why a well-understood idea in machine learning remains apparently so severely underexploited in recommender system research. When it comes to questioning in the assumption of data greed in recommender systems, it appears that ‘someone else is *not* doing it’, and that more effort needs to be made to move the community towards minimal necessary data.

Writing this paper gave us a direct experience of how easy it is to overlook the wider implications of data use. A reviewer pointed out that the NetFlix data set, used here, has been removed from public availability, citing its deanonymizability. Ironically, this consideration escaped us during our experimentation. We must count ourselves among the researchers who face the challenge of understanding the full implications of a commitment to best practices including minimal necessary data.

In sum, we argue that recommender system research must look at how much data is really necessary to accomplish a given recommendation task. However, we find that moving towards minimal necessary data represents a relatively small change in current practices. Recommender system researchers have acquired years of experience addressing cold start. It is time to shift our perspective to realize that cold start is not only a problem, it is also a solution.

7 ACKNOWLEDGEMENTS

This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

REFERENCES

- [1] The data economy: Fuel of the future. *The Economist*, 6–12 May, pages 13–16, 2017.
- [2] F. Abel, A. Benczúr, D. Kohlsdorf, M. Larson, and R. Pálóvics. RecSys Challenge 2016: Job recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys ’16, pages 425–426, 2016.
- [3] J. Bennett, S. Lanning, et al. The Netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA, 2007.
- [4] P. Campos, F. Diez, and I. Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. 24(1-2):67–119, 2014.
- [5] P. G. Campos, A. Bellogin, F. Diez, and J. E. Chavarriaga. Simple time-biased KNN-based recommendations. In *Proceedings of the Workshop on Context-Aware Movie Recommendation*, CAMRa ’10, pages 20–23, 2010.
- [6] R. Chow, H. Jin, B. Knijnenburg, and G. Saldamli. Differential data analysis for recommender systems. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys ’13, pages 323–326, 2013.
- [7] P. Cremonesi, F. Garzotto, and R. Turrin. Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study. *ACM Trans. Interact. Intell. Syst.*, 2(2):11:1–11:41, June 2012.
- [8] P. Cremonesi and R. Turrin. Analysis of cold-start recommendations in IPTV systems. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys ’09, pages 233–236, 2009.
- [9] P. Cremonesi and R. Turrin. Time-evolution of IPTV recommender systems. In *Proceedings of the 8th European Conference on Interactive TV and Video*, EuroITV ’10, pages 105–114, 2010.
- [10] M. D. Ekstrand, F. M. Harper, M. C. Willemsen, and J. A. Konstan. User perception of differences in recommender algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys ’14, pages 161–168, 2014.
- [11] G. Forman and I. Cohen. *Learning from Little: Comparison of Classifiers Given Little Training*, pages 161–172. Springer Berlin Heidelberg, 2004.
- [12] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: a free recommender system library. RecSys ’11. ACM, 2011.
- [13] S. Gordea and M. Zanker. Time filtering for better recommendations with small and sparse rating matrices. In *Proceedings of the 8th International Conference on Web Information Systems Engineering*, WISE ’07.
- [14] F. M. Harper and J. A. Konstan. The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, Dec. 2015.
- [15] A. Karatzoglou. Collaborative temporal order modeling. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys ’11, pages 313–316, 2011.
- [16] B. Kille, A. Lommatzsch, F. Hopfgartner, M. Larson, and A. P. de Vries. A stream-based resource for multi-dimensional evaluation of recommender algorithms. SIGIR ’17 to appear, 2017.
- [17] Y. Koren. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53(4):89–97, Apr. 2010.
- [18] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.
- [19] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), 2007.
- [20] B. Loni and A. Said. Wraprec: An easy extension of recommender system libraries. In *Proceedings of the 8th ACM Conference on Recommender Systems*, RecSys ’14, pages 377–378, 2014.
- [21] V. Maidel, P. Shoval, B. Shapira, and M. Taieb-Maimon. Evaluation of an ontology-content based filtering method for a personalized newspaper. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys ’08, pages 91–98, 2008.
- [22] E. Minkov, B. Charrow, J. Ledlie, S. Teller, and T. Jaakkola. Collaborative future event recommendation. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM ’10, pages 819–828, 2010.
- [23] R. Pagano, P. Cremonesi, M. Larson, B. Hidasi, D. Tikk, A. Karatzoglou, and M. Quadrana. The contextual turn: From context-aware to context-driven recommender systems. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys ’16, 2016.
- [24] S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3), May 2012.
- [25] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI ’09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.
- [26] A. Said, D. Tikk, K. Stumpf, Y. Shi, M. A. Larson, and P. Cremonesi. Recommender systems evaluation: A 3D benchmark. In *ACM RecSys 2012 Workshop on Recommendation Utility Evaluation: Beyond RMSE*, 2012.
- [27] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Investigation of various matrix factorization methods for large recommender systems. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, NETFLIX ’08, pages 6:1–6:8, 2008.

On Quantifying Knowledge Segregation in Society

Abhijnan Chakraborty^{*,*}, Muhammad Ali[#], Saptarshi Ghosh^{*},
Niloy Ganguly^{*}, Krishna P. Gummadi[#]

[#]Max Planck Institute for Software Systems, Germany

^{*}Indian Institute of Technology Kharagpur, India

ABSTRACT

With rapid increase in online information consumption, especially via social media sites, there have been concerns on whether people are getting *selective exposure* to a biased subset of the information space, where a user is receiving more of what she already know, and thereby potentially getting trapped in *echo chambers* or *filter bubbles*. Even though such concerns are being debated for some time, it is not clear how to quantify such echo chamber effect. In this position paper, we introduce *Information Segregation* measures, which follow the long lines of work on residential segregation. We believe that information segregation nicely captures the notion of exposure to different information by different population in a society, and would help in quantifying the extent of social media sites offering selective (or diverse) information to their users.

1 INTRODUCTION

As increasing number of users are consuming information online, often via social media sites like Facebook and Twitter, there have been concerns regarding content quality [6], and the possibility of *biases* in the information people are getting exposed to [3–5, 7]. In such sites, people tend to be connected with other like-minded users out of homophily [1], and thus individual users can have *selective exposure* to information which closely matches their own views, and may not have enough exposure to differing views. There have been further concerns over the effect of such *echo chambers* [7] on the *polarization* of society [8, 13].

Interestingly, in past works, two competing theories of opinion polarization have been proposed [12]. One school of thought assumes that opinions are reinforced when likeminded individuals interact with each other [8, 13]. Whereas, other researchers have argued that exposure to differing views and their subsequent rejections lead to polarization [2]. Polarization can be thought as a *measure of the ideological state* of the population in a society, which is difficult to quantify in general. Also, it is not explicitly clear what constitutes the ideal notion of the *depolarized* state of a society.

In this position paper, we argue that an alternative option would be to consider the access to different types of information by members of a society. For example, within a population with multiple parties operating, it is but natural that political opinion would be fragmented. However, it is highly desirable that the entire population have access to the same information / knowledge and they take informed decision to follow different paths. In other words, the bigger issue here is *whether different groups of people are having*



Figure 1: Basis for computing residential segregation: bipartite matching between people and residential units in a city.

access to similar kind of information or not, where groups may be formed based on predefined demographics (e.g., gender, race, age, income level) or derived features (e.g., political leaning) of people.

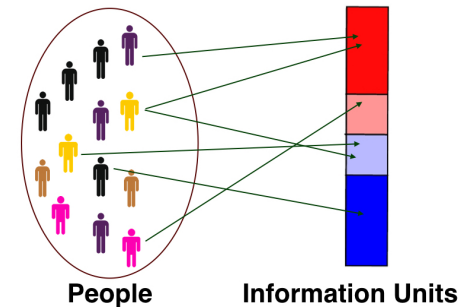


Figure 2: Basis for computing information segregation: bipartite matching between people and information units.

To investigate this issue, we borrow ideas from the past literature on *residential segregation*. A large number of research works have considered the bipartite matching between different groups of people and the urban units where they reside (as shown in Figure 1), and proposed different measures to quantify geographical segregation of different groups [9, 10]. Massey and Denton [14] identified five distinct dimensions of residential segregation:

- (i) **Evenness** is the degree to which groups are distributed proportionately across areal units in an urban area.
- (ii) **Exposure** is the extent to which members of different groups share common residential areas.
- (iii) **Concentration** refers to the degree of a group's agglomeration in urban space.
- (iv) **Centralization** is the extent to which group members reside towards the center of an urban area, and
- (v) **Clustering** measures the degree to which different groups are located adjacent to one another.

Then, they grouped different segregation measures along these five dimensions. Note that some segregation measures are *relative*

between two groups, whereas others are *absolute measures of the segregation of one particular group*.

Following this line of work, in this paper, we present the notion of **Information (or Informational) Segregation**. Similar to Figure 1, we consider another bipartite matching between different groups of people and the information units they have access to (shown in Figure 2). Then utilizing this mapping, we can compute information segregation to measure whether different groups in a society are having access to similar kind of information or not.

However, there are two primary aspects where the mapping between people and information units differs from the mapping between people and residential units: (i) residential segregation is computed over a two-dimensional geographical space, whereas information segregation needs to be computed over a n -dimensional topic space ($n = 1$ in Figure 2, but in general, $n \geq 1$), and (ii) one person may have access to multiple information units, which needs to be accounted for while computing information segregation; whereas, one person is considered to be permanently staying in only one residential unit. To account for people accessing different information units, we use the notion of **fractional personhood** [15]. For an information unit i , we consider the personhood of 1 for everyone who have access to only i , personhood of $\frac{1}{2}$ for them who have access to i and another information unit, and so on.

In this paper, we propose five measures of information segregation analogous to the residential segregation measures discussed earlier, by considering the fractional personhoods of people from different groups. Then, as a proof of concept, we measure the information segregation of US-based Facebook users as evident from how they follow different news media pages on Facebook. Our investigation reveals that Hispanic users are accessing information more evenly across political spectrum; whereas Asian Americans have highest information segregation among all racial groups. Similarly, we also looked at how users having different political leanings are accessing contrary views. We found that moderately conservative leaning users tend to get information more evenly across the spectrum; whereas, extremely conservative leaning users are most segregated among others.

The information segregation measures proposed in this paper can also be used to evaluate the role of search / recommender systems for exposing different types of information to a large population. We believe that in future, greater emphasis should be put on designing more responsible search / recommender systems which limit information segregation to acceptable limits.

2 INFORMATION SEGREGATION MEASURES

In this section, we introduce different measures of information segregation, considering the five distinct dimensions as identified by Messey and Denton [14] for residential segregation.

I. Evenness

The evenness measure of information segregation captures how uniformly members of a particular group have access to different units in the n -dimensional information space. Figure 3 shows an example scenario where members of Yellow group have access to all four information units; whereas, members of Purple group have access to only two units. Therefore, Yellow group in Figure 3 have more even information access than Purple group. Massey and Denton [14] discussed five different measures of residential evenness (including

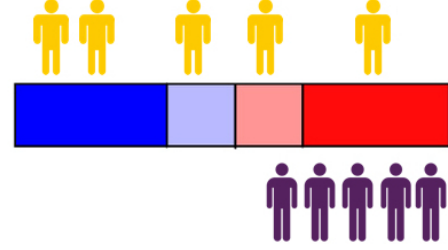


Figure 3: Yellow group gets information more evenly than Purple group.

both relative and absolute measures). For brevity, we are defining only one measure of absolute evenness of a group, which is the complement of **Gini Coefficient** [9].

Gini coefficient G_A measures the unevenness of a particular group A , by capturing the mean absolute difference between the personhoods of A having access to different information units. Then, Information Evenness IE_A can be computed as

$$IE_A = 1 - G_A = 1 - \frac{\sum_{i=1}^m \sum_{j=1, j \neq i}^m |a_i - a_j|}{2 \cdot a_{total} \cdot a'_{total}}$$

where a_i is the sum of personhoods belonging to group A who get information i , a_{total} is the size of group A in the overall population, m is the number of information units, and a'_{total} is the number of people in the overall population who *do not belong* to group A . IE_A varies between 0 to 1, higher the value, the group has more even information access.

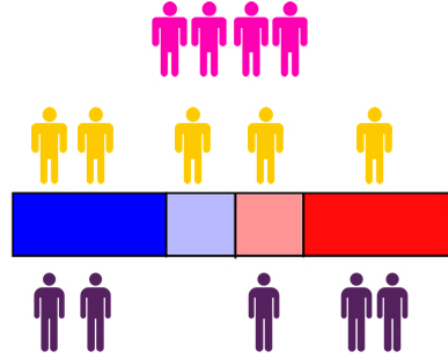


Figure 4: Joint exposure between Purple and Yellow group is higher compared to Purple and Pink group.

II. Joint Exposure

Joint exposure quantifies the extent to which members of two groups get jointly exposed to the same information. In Figure 4, members of Purple and Yellow groups are jointly exposed to three out of four information units; whereas, members of Purple and Pink groups are jointly exposed to only one unit. Therefore, in Figure 4, Purple and Yellow groups have higher joint exposure compared to Purple and Pink groups.

Again using the notion of personhoods, joint information exposure between groups A and B is computed as

$$JIE_{AB} = \sum_{i=1}^m \frac{a_i}{a_{total}} \cdot \frac{b_i}{total_i}$$

where a_i , a_{total} , and m are as defined earlier, b_i is sum of personhoods belonging to B who get information i , and $total_i$ is sum of all personhoods having access to information i . JIE_{AB} varies between 0 to 1, higher the value, A and B have more common exposure.

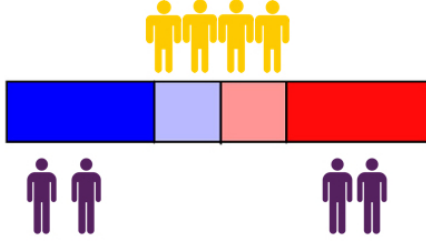


Figure 5: Yellow is more concentrated than Purple group.

III. Concentration

Concentration of a group A refers to the relative amount of topical space that A have access to. Every information unit may not have similar topical density (or number of information sources, etc), with some units having more topics mapping into it, compared to other information units. For example, in Figure 5, red and blue units consist of higher number of topics than blueish and reddish grey units. Therefore, even though Yellow and Purple groups have access to same number of units (hence have same evenness), Yellow group would be considered more concentrated (i.e., more segregated) as it has access to fewer topics. Information concentration is captured by the metric **Delta** [11]:

$$DEL_A = \frac{1}{2} \sum_{i=1}^m \frac{a_i}{a_{total}} \cdot \frac{n_i}{n_{total}}$$

where a_i , a_{total} , and m are already defined, n_i is number of topics in information unit i , and n_{total} is number of topics overall.

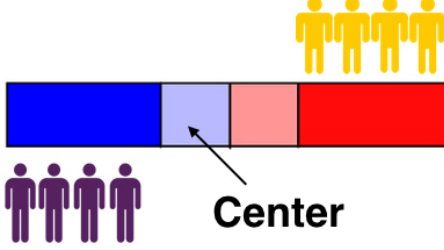


Figure 6: Purple is more centralized than Yellow group.

IV. Centralization

Compared to the geographical context, identifying the center of an information space is tricky, and may not be always possible. Centrality may be computed by considering centroids in a dimension-reduced topical space, or by measuring it over networks induced by information units and their topical or preference similarity. In scenarios where the notion of information center is defined, *centralization between two groups A and B refers to how the information units that A and B have access to are distributed around the center*. For example, in Figure 6, if we assume the blueish grey unit to be the center, then although Yellow and Purple groups have same evenness and concentration measures, Purple group is more centralized than Yellow group. Formally, **Centralization Index** [10] can be measured as

$$CI_{AB} = \sum_{i=1}^m a_{i-1} b_i - \sum_{i=1}^m a_i b_{i-1}$$

where information units are sorted based on their distance from the center, and a_i , b_i , and m are as defined earlier. CI_{AB} varies between -1 to 1 , positive value indicating A is more centralized than B .

V. Clustering

The final dimension of information segregation is the degree to

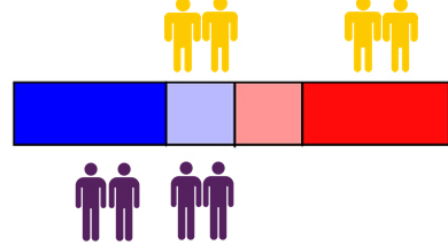


Figure 7: Purple group is more clustered than Yellow group.

which members of a group A have access to information clusters, i.e., whether the different types of information received by A are close to each other in the information space. In Figure 7, both Purple and Yellow groups have access to two information units, and have the same evenness and concentration scores. However, as the information units Purple group have access to are close to each other, according to clustering measure, it is more segregated than Yellow group. We can formally define information clustering as

$$IC_A = \frac{(\sum_{i=1}^m \frac{a_i}{a_{total}} \sum_{j=1}^m e^{-d_{ij}} a_j) - (\frac{a_{total}}{m^2} \sum_{i=1}^m \sum_{j=1}^m e^{-d_{ij}})}{(\sum_{i=1}^m \frac{a_i}{a_{total}} \sum_{j=1}^m e^{-d_{ij}} total_j) - (\frac{a_{total}}{m^2} \sum_{i=1}^m \sum_{j=1}^m e^{-d_{ij}})}$$

where a_i , a_{total} , $total_j$, and m are as defined earlier, and d_{ij} is the distance between information units i and j . IC_A varies from 0 to 1.

3 INFORMATION SEGREGATION AMONG US-BASED FACEBOOK USERS

Next, we attempt to quantify information segregation of Facebook users in the US. Towards that end, we specifically focus on news media pages in Facebook, and measure information segregation with respect to how different groups of users follow these pages.

Dataset Gathered

We queried Facebook search with the term ‘US news media’ to collect US related news media pages in Facebook, and found more than 2.5K Facebook pages for that query. Then using Facebook’s ad submission web page (facebook.com/ads/manager/creation), we collected the composition of gender, race and political leanings of the followers of these media pages. We acknowledge the limitation that the retrieved pages may not be representative of all US media pages, and we would expand the corpus in future work.

Mapping Facebook Pages to Information Units

To quantify information segregation, we focus on 1-dimensional political information space, and divide it into five information units: **Very Conservative (VC)**, **Conservative (C)**, **Moderate (M)**, **Liberal (L)**, and **Very Liberal (VL)**. Then, we map different news pages on Facebook to one of these five information units by considering the political leanings of the followers of these pages. For a page P , if the fraction of followers leaning towards respective political ideologies are denoted as f_{VC} , f_C , f_M , f_L , and f_{VL} respectively, then we measure the political leaning of P ($Leaning_P$) as a weighted sum of the political leaning of its audience. More specifically,

$$Leaning_P = -1 \cdot f_{VC} + -0.5 \cdot f_C + 0 \cdot f_M + 0.5 \cdot f_L + 1 \cdot f_{VL}$$

If $Leaning_P$ is between -0.1 to $+0.1$, we map P to information unit M ; for $Leaning_P$ between 0.1 to 0.5 , P is mapped to L and for $Leaning_P > 0.5$, we map P to VL . Similarly, we map P to C or VC if $-0.5 \leq Leaning_P < -0.1$ and $Leaning_P < -0.5$ respectively.

Computing the Personhood Scores

After mapping every page to one of the information units, we try

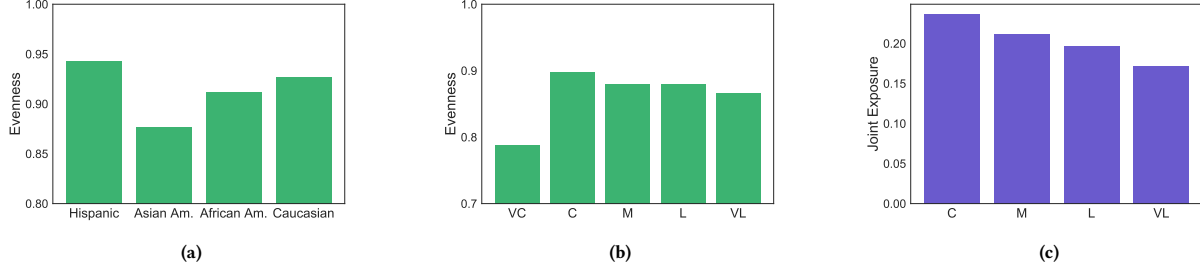


Figure 8: Information segregation between different groups along two dimensions: evenness of (a) different racial groups, (b) different political groups, and (c) joint exposure of very conservative leaning people (VC) with other political groups.

to gather the cumulative number of followers for a particular unit. However, Facebook doesn't allow us to get the follower size for a combination of more than 400 Facebook pages. Therefore, we randomly select 400 pages from the set of 2.5K+ news media pages, map them to their corresponding units, and gather the demographics of the followers of pages belonging to every information unit.

As some users may follow Facebook pages belonging to multiple units (for example, follow both conservative and liberal leaning pages), we need to accurately account for these overlaps in information access. As mentioned earlier, we use the notion of *fractional personhood* in this regard. Therefore, instead of considering the number of followers of pages in a particular unit, we consider the sum of personhoods for pages in every information unit.

For every unit i , the sum of personhoods N_i^* is computed as

$$N_i^* = [N(S) - N(S \setminus i)] + \frac{1}{2} \sum_{j \in (S \setminus i)} N(i \cap j) + \frac{1}{3} \sum_{j \in (S \setminus i)} \sum_{k \in (S \setminus i \setminus j)} N(i \cap j \cap k) + \dots$$

where S is the set of all information units $\{VC, C, M, L, VL\}$ and $N(x)$ gives the number of followers of pages in unit(s) x .

Information Segregation among Racial Groups

Facebook ad interface returns four racial categories for the users: **Caucasian, African American, Asian American, and Hispanic**. For every information unit, we compute the personhoods belonging to each race, and then measure information segregation among them. Figure 8(a) shows the evenness of different racial groups. We can see in Figure 8(a) that Hispanics have most even access to different political information units; whereas, Asian Americans have most uneven access to political information units.

Information Segregation between Political Groups

Similar to the racial categories, we also computed the personhoods w.r.t. different political leanings for every information unit, and then measure the information segregation among these groups. Figure 8(b) shows that conservative leaning users tend to get information evenly from information units; whereas, very conservative leaning users have most uneven access to different units. Then to measure how very conservative leaning users have common access to information units with others, we plot their joint exposure with other groups in Figure 8(c). We observe that very conservative leaning users have highest joint exposure with conservatives, denoting that they are exposed to multiple information units together. Whereas, they have least joint exposure with very liberal leaning users, implying that these two groups have access to very different information units.

4 CONCLUSION

In this position paper, we proposed five measures of information segregation motivated by the residential segregation measures proposed in literature. Then, using these measures, we computed information segregation among US-based Facebook users. Our future work lies in evaluating how search / recommender systems are exposing information to different groups of users, and proposing mechanisms to keep information segregation to acceptable limits.

Acknowledgments: The authors thank the anonymous reviewers whose suggestions helped to improve the paper. A. Chakraborty is a recipient of Google India PhD Fellowship and Prime Minister's Fellowship Scheme for Doctoral Research, a public-private partnership between Science & Engineering Research Board (SERB), Department of Science & Technology, Government of India and Confederation of Indian Industry (CII).

REFERENCES

- [1] Luca Maria Aiello, Alain Barrat, Rossano Schifanella, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. 2012. Friendship prediction and homophily in social media. *ACM TWEB* 6, 2 (2012).
- [2] Delia Baldassarri and Andrew Gelman. 2008. Partisans without Constraint: Political Polarization and Trends in American Public Opinion. *Sociology* (2008).
- [3] Abhijnan Chakraborty, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. 2015. Can trending news stories create coverage bias? on the impact of high content churn in online news media. In *Computation and Journalism Symposium*.
- [4] Abhijnan Chakraborty, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. 2016. Dissemination Biases of Social Media Channels: On the Topical Coverage of Socially Shared News. In *AAAI ICWSM*.
- [5] Abhijnan Chakraborty, Johnatan Messias, Fabrizio Benevenuto, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. 2017. Who Makes Trends? Understanding Demographic Biases in Crowdsourced Recommendations. In *AAAI ICWSM*.
- [6] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. In *IEEE/ACM ASONAM*.
- [7] Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. 2014. Echo chamber or public sphere? Predicting political orientation and measuring political homophily in Twitter using big data. *Journal of Communication* (2014).
- [8] Pranav Dandekar, Ashish Goel, and David T Lee. 2013. Biased assimilation, homophily, and the dynamics of polarization. *PNAS* (2013).
- [9] Robert Dorfman. 1979. A formula for the Gini coefficient. *Review of Economics and Statistics* (1979).
- [10] Otis Dudley Duncan and Beverly Duncan. 1955. Residential distribution and occupational stratification. *Am. journal of sociology* (1955).
- [11] Edgar M Hoover. 1941. Interstate redistribution of population, 1850–1940. *Journal of Economic History* (1941).
- [12] Michael Maes and Lukas Bischofberger. 2015. Will the Personalization of Online Social Networks Foster Opinion Polarization? (2015).
- [13] Michael Maes and Andreas Flache. 2013. Differentiation without distancing. Explaining opinion bi-polarization without assuming negative influence. *Plos One* (2013).
- [14] Douglas S Massey and Nancy A Denton. 1993. *American apartheid: Segregation and the making of the underclass*.
- [15] Christian Perring. 1997. Degrees of personhood. *Medicine and Philosophy* (1997).

Balanced Neighborhoods for Fairness-aware Collaborative Recommendation

Robin Burke
School of Computing
DePaul University
Chicago, Illinois
rburke@cs.depaul.edu

Masoud Mansoury
School of Computing
DePaul University
Chicago, Illinois
mmansou4@depaul.edu

Nasim Sonboli
School of Computing
DePaul University
Chicago, Illinois
nsonboli@depaul.edu

Aldo Ordoñez-Gauger
School of Computing
DePaul University
Chicago, Illinois
aordone3@mail.depaul.edu

ABSTRACT

Recent work on fairness in machine learning has begun to be extended to recommender systems. While there is a tension between the goals of fairness and of personalization, there are contexts in which a global evaluation of outcomes is possible and where equity across such outcomes is a desirable goal. In this paper, we introduce the concept of a balanced neighborhood as a mechanism to preserve personalization in recommendation while enhancing the fairness of recommendation outcomes. We show that a modified version of the SLIM algorithm can be used to improve the balance of user neighborhoods, with the result of achieving greater outcome fairness in a real-world dataset with minimal loss in ranking performance.

KEYWORDS

Fairness, Recommender systems, Machine Learning, Neighborhood Models

ACM Reference format:

Robin Burke, Nasim Sonboli, Masoud Mansoury, and Aldo Ordoñez-Gauger. 2017. Balanced Neighborhoods for Fairness-aware Collaborative Recommendation. In *Proceedings of ACM FATRec Workshop, Como, Italy, August 2017 (FATRec'17)*, 5 pages.
<https://doi.org/10.18122/B2GQ53>

1 INTRODUCTION

Bias and fairness in machine learning are topics of considerable recent research interest [1, 3]. A standard approach in this area is to identify a variable or variables representing membership in a protected class, for example, race in an employment context, and to develop algorithms that remove bias relative to this variable. See, for example, [7, 8, 11, 13, 14].

To extend this concept to recommender systems, we must recognize the key role of personalization. Inherent in the idea of recommendation is that the best items for one user may be different than those for another. The dominant recommendation paradigm, collaborative filtering [9], uses user behavior as its input, ignoring

user demographics and item attributes. One approach to fairness in recommendation is to examine outcomes only in terms of the level and type of error experienced by different groups [12]. However, there are contexts in which this approach may be insufficient. Consider a recommender system suggesting job opportunities to job seekers. An operator of such a system might wish, for example, to ensure that male and female users with similar qualifications get recommendations of jobs with similar rank and salary. The system would therefore need to defend against biases in recommendation output, even biases that might arise entirely due to behavioral differences: for example, male users might be more likely to click optimistically on high-paying jobs.

Defeating such biases is difficult if we cannot assert a shared global preference ranking over items. Personal preference is the essence of recommendation especially in areas like music, books, and movies where individual taste is paramount. Even in the employment domain, some users might prefer a somewhat lower-paying job if it had other advantages: such as flexible hours, shorter commute time, or better benefits. Thus, to achieve the policy goal of fair recommendation of jobs by salary, a site operator must go beyond personalization as a goal and impose additional constraints on the recommendation algorithm.

In this paper, we investigate fairness-aware recommendation in the context of recommendation. In particular, we develop the idea of segregation in recommendation, its implications for fairness, and show that a regularization-based approach can be used to control the formation of recommendation neighborhoods. We show that this approach can be used to overcome statistical biases in the distribution of recommendations across users in different groups.

2 BALANCED NEIGHBORHOODS IN RECOMMENDATION

In [13], the authors impose a fairness constraint on a classification by creating a *fair representation*, a set of prototypes to which instances are mapped. The prototypes each have an equal representation of users in the protected and unprotected class so that the association between an instance and a prototype carries no information about the protected attribute.

This article may be copied, reproduced, and shared under the terms of the Creative Commons Attribution-ShareAlike license (CC BY-SA 4.0).

FATRec'17, August 2017, Como, Italy

© 2017 Copyright held by the owner/author(s).

DOI: 10.18122/B2GQ53

As noted above, the requirement for personalization in recommendation means that we have as many classification tasks as we have users. A direct application of the fair prototype idea would aggregate many users together and produce the same recommendations for all, greatly reducing the level of personalization and the recommendation accuracy. This idea must be adapted to apply to recommendation.

One of the fundamental ideas of collaborative recommendation is that of the *peer user*, a neighbor whose patterns of interest match those of the target user and whose ratings can be extrapolated to make recommendations for the target user. One place where bias may creep into collaborative recommendation may be through the formation of peer neighborhoods.

Consider the situation in Figure 1. The target user here is the solid square, a member of the protected class. The top of the figure shows a neighborhood for this user in which recommendation will be generated only from other square users, that is, other protected individuals. We can think of this as a kind of segregation of the recommendation space. If the peer neighborhoods have this kind of structure relative to the protected class, then this group of users will only get recommendations based on the behavior and experiences of users in their own group. For example, in the job recommendation example above, women would only get recommendations of jobs that have interested other women applicants, potentially leading to very different recommendation experiences across genders.

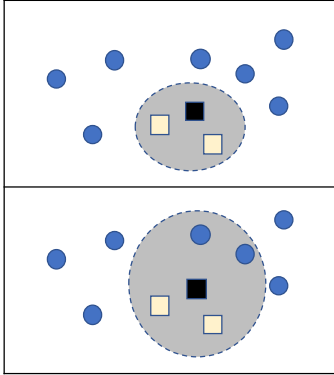


Figure 1: Unbalanced (top) and balanced (bottom) neighborhoods

To counter this type of bias, we introduce the notion of a *balanced neighborhood*. A balanced neighborhood is one in which recommendations for all users are generated from neighborhoods that are balanced with respect to the protected and unprotected classes. This is shown in the bottom half of Figure 1. The target has an equal number of peers inside and outside of the protected class. In the case of job recommendation discussed above, this would mean that female job seekers get recommendations from some female and some male peers.

There are a variety of ways that balanced neighborhoods might be formed. The simplest way would be to create neighborhoods for each user that balance accuracy against group membership. This could be highly computationally inefficient as it would require solving a separate optimization problem for each user. In this research,

we explore an extension of the well-known Sparse Linear Method (SLIM), which has been proved very effective in recommendation ranking with implicit data. This extension uses regularization to control the way different neighbors are weighted, with the goal of achieving balance between protected and non-protected neighbors for each user.

3 SLIM

The Sparse Linear Method for recommendation was introduced in [10]. It is a generalization of item-based k-nearest neighbor in which all items are used and weights for these items are learned through optimization to minimize a regularized loss function. Although this is not proposed in the original SLIM paper, it is possible to create a user-based version of SLIM (labeled SLIM-U in [15]), which generalizes the user-based algorithm in the same way.

Assume that there are M users (a set U), N items (a set I), and let us denote the associated 2-dimensional rating matrix by R . SLIM is designed for item ranking and therefore R is typically binary. We will relax that requirement in this work. We use u_i to denote user i and t_j to denote the item j . An entry, r_{ij} , in matrix R represents the rating of u_i on t_j .

SLIM-U predicts the ranking score \hat{s} for a given user, item pair $\langle u_i, t_j \rangle$ as a weighted sum:

$$\hat{s}_{ij} = \sum_{k \in U} w_{ik} r_{kj}, \quad (1)$$

where $w_{ii} = 0$ and $w_{ik} \geq 0$.

Alternatively, this can be expressed as a matrix operation yielding the entire prediction matrix \hat{S} :

$$\hat{S} = WR, \quad (2)$$

where W is an $M \times M$ matrix of user-user weights. For efficiency, it is very important that this matrix be sparse.

The optimal weights for SLIM-U can be derived by solving the following minimization problem:

$$\min_W \frac{1}{2} \|R - WR\|^2 + \lambda_1 \|W\|^1 + \frac{\lambda_2}{2} \|W\|^2, \quad (3)$$

subject to $W > 0$ and $\text{diag}(W) = 0$.

The $\|W\|^2$ term represents the ℓ_2 norm of the W matrix and $\|W\|^1$ represents the ℓ_1 norm. These regularization terms are present to constrain the optimization to prefer sparse sets of weights. Typically, coordinate descent is used for optimization. Refer to [10] for additional details.

3.1 Neighborhood Balance

Recall that our aim in fair recommendation is to eliminate segregated recommendation neighborhoods where protected class users only receive recommendations from other users in the same class. Such neighborhoods would tend to magnify any biases present in the system: if users in the protected class only are recommended certain items, then they will be more likely to click on those items and thus increase the likelihood that the collaborative system will make these items the ones that others in the protected group see.

To reduce the probability that such neighborhoods will form, we use the SLIM-U formalization of the recommendation problem, but we add another regularization term to the loss function, which we

call the *neighborhood balance* term. To describe this term, we will enrich our notation further by indicating U^+ to be the subset of U containing users in the protected class with the remaining users in the class U^- . Let W_i^+ be the set of weights for users in U^+ and W_i^- be the corresponding set of weights for the non-protected class. Then the neighborhood balance term b_i for a given user i is the squared difference between the weights assigned to peers in the protected class versus the unprotected class.

$$b_i = \left(\sum_{w^+ \in W_i^+} w^+ - \sum_{w^- \in W_i^-} w^- \right)^2 \quad (4)$$

A low value for the neighborhood balance term means that the user's predictions will be generated by weighting protected and unprotected users on a relatively equal basis.

Note that this is a class-blind optimization that tries to build balanced neighborhoods for both the protected and unprotected users. It is also possible to formulate the objective such that it only impacts the protected class and we will leave this option for future work. If the classes are highly imbalanced, it may be necessary to weight these terms so that the weights are expected to sum in proportion to the size of each group. We will explore this idea in future work.

Another way to express this idea is to create a vector p of dimension M . If u_i is in U^+ , then $p_i = 1$; if u_i is in U^- , then $p_i = -1$. Then, the sum expressed above can be rewritten as $b_i = \|p^T w_i\|^2$. By adding up this term for all users and adding it to the loss function, we can allow the optimization process to derive weights with neighborhood balance in mind. This adapted version of SLIM-U we will call *Balanced Neighborhood SLIM* or BN-SLIM.

As in the case of SLIM, we can apply the method of coordinate descent to optimize the objective. The basic algorithm is to choose one w_{ik} weight and solve the optimization problem for that weight, repeating over all the weights until convergence is reached. The full loss function is as follows:

$$L = \frac{1}{2} \|R - WR\|^2 + \lambda_1 \|W\|^1 + \frac{\lambda_2}{2} \|W\|^2 + \frac{\lambda_3}{2} \sum_{i \in U} \left(\sum_{k \in U} p_i w_{ik} \right)^2, \quad (5)$$

where $w_{ii} = 0$ and $w_{ik} \geq 0$ and where λ_3 is a parameter controlling the influence of the neighborhood balance calculation on the overall optimization

This loss function retains the property of the original SLIM algorithm in that the rows of the weight matrix are independent, and the weights in each row (those for each user) can be optimized independently. If we take the derivative of L with respect to a single weight w_{ik} , we obtain

$$\frac{\partial L_i}{\partial w_{ik}} = \sum_{j \in I} (r_{ij} - \sum_{l \in U'} w_{il} r_{lj}) + w_{ik} \sum_{j \in I} r_{kj}^2 + \lambda_1 + \lambda_2 w_{ik} + \lambda_3 p_k \sum_{l \in U'} p_l w_{il} \quad (6)$$

where $U' = U - \{u_i, u_k\}$.

We then set this derivative to zero and solve for the value of w_{ik} that produces this minimum. This becomes the coordinate descent update step.

$$w_{ik} \leftarrow \frac{S \left(\sum_{j \in I} (r_{ij} - \sum_{l \in U'} w_{il} r_{lj}) + \lambda_3 p_k \sum_{l \in U'} p_l w_{il}, \lambda_1 \right)_+}{\sum_{j \in I} r_{kj}^2 + \lambda_2 + \lambda_3} \quad (7)$$

where $S(\cdot)_+$ is the soft threshold operator defined in [4].

4 METHODOLOGY

It is very difficult to find datasets that contain the kind of features that would be necessary to evaluate fairness-aware recommendation algorithms based on user demographics. For example, the data from the job search site XING¹ that was made available for the 2017 RecSys Challenge² does not have any demographic information about users except their broad geographic region.

For the purposes of demonstration, we are using the MovieLens 1M dataset [6], which contains user gender information. Movie recommendation is, of course, a domain of pure individual taste and therefore not an obvious candidate for fairness-aware recommendation. Following the example of [12], our approach to construct an artificial equity scenario within this data for expository purposes only, with the understanding that real scenarios can be approached with a similar methodology.

Our artificial scenario centers on movie genres. It can be seen in this data that there is a minority of female users (1709 out of the total of 6040). Certain genres display a discrepancy in recommendation delivery to male and female users. For example, in the "Crime" genre, female users rate a very similar number of movies (average of 0.048% of female profiles vs 0.049% of male profiles) and rate them similarly: an average rating of 3.689 for female users vs 3.714 for male users. However, our baseline unmodified SLIM-U algorithm recommends in the top 10 an average of 1.10 "Crime" movies per female user as opposed to 1.18 such movies to male users. We are still exploring the cause of this discrepancy, but it seems likely that there are influential female users with a lower opinion of this genre.

Given that the rating profiles are similar but the recommendation outcomes are different, we can therefore conclude that the female users experience a deprivation (if one wants to call it that) of "Crime" movies compared to their male counter-parts. Similar losses can be observed for other genres. It is, of course, questionable if there is any harm associated with this outcome and we do not claim such. It is sufficient that these differences allow us to validate the properties of the BN-SLIM algorithm.

Our goal, then, is to reduce or eliminate genre discrepancies with minimal accuracy loss by constructing balanced neighborhoods for the MovieLens users. The p vector in Equation 7 therefore will have a 1 for female users and a -1 for male users. In the experiments below, we compare the user-based SLIM algorithm in its unmodified form and the balanced neighborhood version BN-SLIM.

In evaluating fairness of outcome, we measure the number of movies of the chosen genre as the measure of outcome quality.

¹<https://www.xing.com/jobs>

²<http://2017.recsyschallenge.com/>

Therefore, we construct a genre-level equity score, $E@k$ for recommendation lists of k items, as the ratio between the outcomes for the different groups. Let $P_i@k = \rho_1, \rho_2, \dots, \rho_k$ be the top k recommendation list for user i , and let $c()$ be a function $\rho \rightarrow 0, 1$ that maps to 1 if the recommended movie is in the chosen genre. Then:

$$E@k = \frac{\sum_{i \in U^+} \sum_{\rho \in P_i@k} c(\rho) / |U^+|}{\sum_{i \in U^-} \sum_{\rho \in P_i@k} c(\rho) / |U^-|} \quad (8)$$

$E@k$ will be less than 1 when the protected group is, on average, recommended fewer movies of the desired genre. It may be unrealistic to imagine that this value should approach 1: the metric does not correct for other factors that might influence this score – for example, female users may rate a particular genre significantly lower and an equality of outcome should not be expected. While the absolute value of the metric may be difficult to interpret, it is still useful for comparing algorithms. The one with the higher $E@k$ is providing more movies in the given genre to the protected group.

As in any multi-criteria setting, we must be concerned about any loss of accuracy that results from taking additional criteria into consideration. Therefore, we also evaluate NDCG@10 for our algorithms in the results below.

5 RESULTS

We implemented the SLIM-U and BN-SLIM algorithms using LibRec 2.0 [5]. We used 5-fold cross-validation as implemented within the library. Within the MovieLens 1M dataset, we selected the five genres on which the SLIM-U algorithm produced the lowest equity scores: “Film-Noir”, “Mystery”, “Horror”, “Documentary”, and “Crime”. The parameters were set as follows: $\lambda_1 = 0.1$, $\lambda_2 = 0.001$, and (for BN-SLIM) $\lambda_3 = 25^3$.

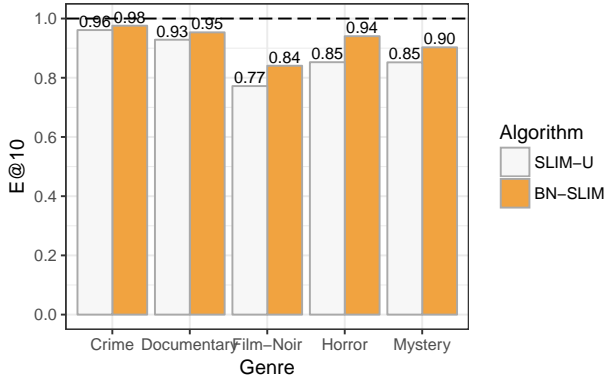


Figure 2: Equity score for SLIM-U and BN-SLIM.
Line indicates equal percentage across genders

Figure 2 shows the results of the experiment in terms of the equity scores for each genre. Perfect equity (1.0) is marked with the dashed line. As we can see, in every case, the balanced neighborhood algorithm produced an equity score closer to 1.0 than the

³Because the balance term measures the difference in weights, it tends to be much smaller than the terms that measure the sums of weights. Therefore, the regularization constant must be much higher for it to have an impact.

Algorithm	NDCG@10
SLIM-U	0.053
BN-SLIM	0.052

Table 1: Ranking accuracy

unmodified algorithm. The largest jump is seen in the “Horror” genre, about 0.09 in the equity score or around 10%.

In terms of accuracy, there was only a small loss of NDCG@10 between the two conditions. See Table 1. The difference amounts to approximately 2% loss in NDCG@10 for the balanced neighborhood version.

Because the balanced neighborhood algorithm is applied across all users, it also has the effect of showing male users movie genres that occur more frequently for female users. To see this effect, we examined the five genres with the highest $E@10$ values: “Fantasy”, “Animation”, “War”, “Romance”, and “Western” using the same parameter values as above. The results appear in Figure 3 and show a similar result. “War” is clearly the anomaly here, both because it is surprising to see it as a one of the more female-recommended genres and because the genre-balance algorithm pushes it to become more skewed rather than less. We are investigating the cause of this phenomenon. Overall, the BN-SLIM algorithm produces a recommendation experience in which the occurrence of gender-specific genres is more closely equalized, with small loss in ranking accuracy.

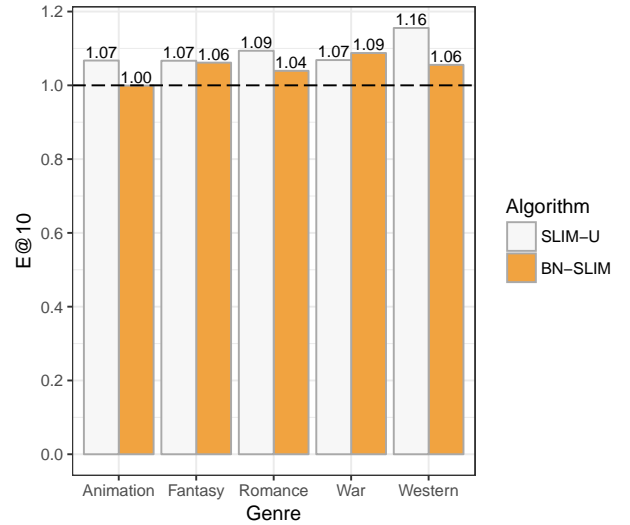


Figure 3: Equity scores for female-preferred genres

6 CONCLUSION

Considerations of fairness and equity are in tension with the focus on personalization that is central to recommender systems research. To ask if a recommendation outcome is fair, by definition, assumes some kind of universal standard for such outcomes, existing outside of individual preference. In some recommendation domains,

such as employment and housing, it is reasonable to expect that recommender systems may be held to such standards.

In this paper, we consider one way in which a fair outcome for a protected group may be sought in the context of personalized recommendation. Drawing on the idea of fair prototypes [13], we propose the construction of balanced neighborhoods as a mechanism for achieving fair outcomes in recommendation and we provide an implementation of the idea using a variant of the Sparse Linear Method.

Although we were not able to demonstrate results in a domain in which fair outcomes are critical, we were able to construct an evaluation using the MovieLens data set and show that our balanced neighborhood implementation overcomes biases inherent in the data with respect to male and female users and the recommendation of different genres with minimal loss in ranking accuracy.

In future work, we hope to acquire appropriate data to evaluate our approach in areas where fairness is of greater societal importance, and to extend the balanced neighborhood approach to other algorithms. Finally, we are also interested in scenarios in which there are fairness considerations for both sides of the recommendation transaction, such as reciprocal recommendation scenarios [2].

7 ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation under grant IIS-1423368.

REFERENCES

- [1] Engin Bozdag. 2013. Bias in algorithmic filtering and personalization. *Ethics and Information Technology* 15, 3 (Sept. 2013), 209–227. <https://doi.org/10.1007/s10676-013-9321-6>
- [2] Burke, Robin. 2017. Multisided Fairness for Recommendation. In *Workshop on Fairness, Accountability and Transparency in Machine Learning (FATML)*. Halifax, Nova Scotia, To appear.
- [3] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 214–226.
- [4] Jerome Friedman, Trevor Hastie, Holger Häußling, and Robert Tibshirani. 2007. Pathwise coordinate optimization. *The Annals of Applied Statistics* 1, 2 (Dec. 2007), 302–332. <https://doi.org/10.1214/07-AOAS131>
- [5] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. LibRec: A Java Library for Recommender Systems.. In *UMAP Workshops*.
- [6] F Maxwell Harper and Joseph A Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4 (2015), 19.
- [7] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination aware decision tree learning. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 869–874.
- [8] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. *Machine Learning and Knowledge Discovery in Databases* (2012), 35–50.
- [9] Y. Koren and R. Bell. 2011. Advances in collaborative filtering. *Recommender Systems Handbook* (2011), 145–186.
- [10] Xia Ning and George Karypis. 2011. Slim: Sparse linear methods for top-n recommender systems. In *11th IEEE International Conference on Data Mining (ICDM)*. IEEE, 497–506.
- [11] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 560–568.
- [12] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. *CoRR* abs/1705.08804 (2017). <http://arxiv.org/abs/1705.08804>
- [13] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. 325–333.
- [14] Lu Zhang and Xintao Wu. 2017. Anti-discrimination learning: a causal modeling-based framework. *International Journal of Data Science and Analytics* (2017), 1–16.
- [15] Yong Zheng, Bamshad Mobasher, and Robin Burke. 2014. CSLIM: Contextual SLIM recommendation algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*. ACM, 301–304.

Impact of Task Recommendation Systems in Crowdsourcing Platforms

Kathrin Borchert, Matthias Hirth
University of Würzburg, Institute of Computer Science
Würzburg, Germany
[kathrin.borchert|matthias.hirth]@informatik.uni-wuerzburg.de

Steffen Schnitzer, Christoph Rensing
University of Darmstadt, Multimedia Communications Lab
Darmstadt, Germany
[steffen.schnitzer|christoph.rensing]@kom.tu-darmstadt.de

ABSTRACT

Commercial crowdsourcing platforms accumulate hundreds of thousand of tasks with a wide range of different rewards, durations, and skill requirements. This makes it difficult for workers to find tasks that match their preferences and their skill set. As a consequence, recommendation systems for matching tasks and workers gain more and more importance. In this work we have a look on how these recommendation systems may influence different fairness aspects for workers like the success rate and the earnings. To draw generalizable conclusions, we use a simple simulation model that allows us to consider different types of crowdsourcing platforms, workers, and tasks in the evaluation. We show that even simple recommendation systems lead to improvements for most platform users. However, our results also indicate and shall raise the awareness that a small fraction of users is also negatively affected by those systems.

ACM Reference format:

Kathrin Borchert, Matthias Hirth and Steffen Schnitzer, Christoph Rensing. 2017. Impact of Task Recommendation Systems in Crowdsourcing Platforms. In *Proceedings of Workshop on Responsible Recommendation at RecSys 2017, Como, Italy, August 2017 (FATREC'17)*, 6 pages. <https://doi.org/10.18122/B2CX1Q>

1 INTRODUCTION

In recent years the diversity of crowdsourcing services and applications has dramatically grown. Especially commercial crowdsourcing platforms focusing on micro tasking, e.g. Amazon Mechanical Turk¹ or Microworkers², accumulate a huge variety of different task types. These tasks, e.g. tagging images or answering surveys, are mostly repetitive, simple and their completion requires only a short amount of time. Regardless of their simplicity, most tasks still need a certain skill set on the worker's side for a successful completion.

The large number and variety of tasks and their individual requirements calls for an automatic solution to help the workers to find suitable tasks which fit their individual interests and capabilities, e.g. by using personalized task recommendation systems. Contributing in such tasks may lead to a higher success rate of the workers and thus ultimately to a higher income. However, it is not

clear if the integration of task recommendation systems in crowdsourcing platforms has solely positive effects. Recommendation systems might also lead to unfairness, as some workers might get assigned to only a small number of tasks or only low paid tasks.

This paper aims at raising the awareness for such potential negative effects of recommendation systems in crowdsourcing platforms. We use a simple simulation model that includes components and processes of a crowdsourcing platform on an abstract level for quantifying and analysing the effects of a task recommendation system. A very basic task recommendation algorithm and a random based approach as baseline are used for the task suggestions. This simple setup allows us to illustrate the benefits and potential drawbacks of recommendation systems in the context of crowdsourcing platforms from a high-level point of view.

The remainder of the paper is structured as followed. The related work in the second section provides an overview of recommendation mechanisms in the context of crowdsourcing. The simulation model is described in the third section, including the models for tasks, workers, recommendation and selection of tasks, as well as the chosen evaluation metrics. The evaluation is presented in the fourth section, where key influence parameters of the simulation model are identified, and a main effect analysis is used to deduce the settings for evaluating the impact of task recommendation for diverse platforms. Further, the evaluation section provides an analysis of the impact of mechanisms on the workers' earning on the diverse platforms. The fifth section concludes the paper with a discussion of the findings.

2 RELATED WORK

Crowdsourcing tasks differ significantly in their complexity and the skills required by the worker completing those tasks [5, 16]. Thus, one possibility to leverage the benefits of recommendation systems in the crowdsourcing context is using them to automatically find suitable tasks for the workers. Several approaches for such task recommendation systems have already been proposed. An overview over different task recommendation approaches and evaluation methods for crowdsourcing in several areas is given by Geiger and Schader [6]. Numerous mechanisms are based on content knowledge, e.g. characteristics of previously completed tasks such as category, reward or allocated time [8, 17]. In addition, Yuen et al. [18] consider the workers interactions, e.g. searching for tasks. Such previous behavior of the workers on the platforms is also used for collaborative filtering algorithms [1, 11]. In contrast to recommending tasks to workers, the concept developed by Difallah et al. [3] realizes a push methodology to find the best suited worker

¹<https://www.mturk.com/> Accessed: Jun. 2017

²<https://www.microworkers.com/> Accessed: Jun. 2017

for a task by extracting interests and skills from an online social network. Still, the evaluation of all these recommendation approaches is limited to the accuracy of the recommendations or the improvement of the quality of the worker input by a practical research or offline experiments. The framework for optimizing task assignment in the field of knowledge intensive tasks introduced by [13] prevents an over or under utilization of the workers but the influence on the involved actors, e.g. reduced earnings, is not investigated.

There are already studies about the disparate impact of algorithms and computational unfairness in several fields [4, 12], e.g. algorithms used in online advertising systems [2]. Further, there are several approaches to overcome the disparate treatment or impact in the area of decision making algorithms [9, 19]. However, to the best of our knowledge, there is no study about the impact of task recommendation systems on the workers in crowdsourcing systems.

3 SIMULATION MODEL

We use a simulation model to evaluate the impact of task recommendation mechanisms in crowdsourcing platforms. In contrast to a real-world implementation in an existing commercial crowdsourcing platform, this allows us to analyse the impact of a broad range of different parameter settings. The remainder of this section gives a brief description of the model components and structure of the implementation. Furthermore, the implementation of the evaluated recommendation algorithms is described. Finally, we introduce the evaluation metrics used to quantify the impact of the task recommendation algorithm on the users.

3.1 Simulation Description

The simulation implements different components of a crowdsourcing platform, such as tasks of various categories, workers and their interactions. Each simulation run is divided into two parts, the initialization of the workers and tasks, and an event based simulation process modelling the interactions of the workers and the platform.

The discrete event simulation is again divided into three steps, the worker selection, the task selection and the task execution. We assume that every idle worker of the worker pool is searching for a task. Thus in the first step (1), we start with the selection of an idle worker. The selection follows a random uniform distribution. In the next step, the task selection (2), the recommendation algorithm determines an available task from the pool of tasks to recommend. In case the recommended task does not fit the worker's skills, with a certain probability the worker selects a suitable task randomly from the task pool by himself. If no such task is available, he accepts the recommendation and starts to work on the selected task. During the task execution step (3), the worker is busy and does not accept other tasks. The duration of the execution process is defined by the required completion time of the task. During this process, the result of the task is computed based on the skills of the worker in the requirements of the task. If the task is successfully completed, the task status will be changed to completed and removed from the system. Otherwise, the task will become available again. At this point one iteration of the event based simulation is completed, the simulation time is updated, and the worker returns to the idle state. The simulation is terminated after a specified time period.

3.2 Simulation Components

In the following we have a closer look at how tasks and workers are represented in the simulation. Models are based on typical structures and characteristics of real micro-tasking platforms, e.g., Amazon Mechanical Turk or Microworkers. Moreover, we explain the implementation of the recommendation algorithm and the used baseline. In the last part of this section we give an overview of the parameters of the simulation model used to specify the characteristics of the simulated platforms.

3.2.1 Task and Category Model. In our model, a *task* requires a set of worker skills to be completed correctly. The required skills are determined by the *category* of the task. Additionally, a task belongs to a *campaign* that groups identical tasks, as they would be submitted by a requester in a real-world platform. A campaign defines the payment, the time required for completion, and the number of identical tasks, as well as the creation time for all of its tasks.

All tasks for one simulation run are created during an initialization phase to optimize the runtime of the simulation. In a first step, m categories are created. Thereafter, the campaigns are generated with negative exponentially distributed inter-arrival times. Negative exponentially distributed inter-arrival times are often a feasible assumption if a large number of traffic sources, or in this case employers, are present. This also allows us to reduce the total number of model parameters, as the higher moments of the arrival process are directly dependent on the mean inter-arrival time, even if other distributions might be more realistic, c.f. [15]. Each campaign is then randomly assigned to a category and the associated campaign properties are added. The last step initializes tasks and adds them to the pool. However, the campaigns and tasks are not directly available at the beginning of the discrete event simulation. During the simulation the state of the tasks is changed to active at the arrival time of the associated campaign.

3.2.2 Worker Model. In our model we assume that there are two basic *worker types*: (1) The specialized worker (*sw*), who prefers tasks of only one category and (2) the average worker (*aw*), who favors multiple categories. The amount of favored categories of the average worker varies between two up to m categories.

Beside the amount of favored categories the worker types differ concerning their skills. The skills are defined by the success probability in each category. The specialized workers *sw* are high skilled in their preferred category. Thus, in their favored categories the success probability p_{sw} is very high. The success probability p_{aw} of the favored categories of average workers is medium, since they do not exclusively focus on one type of tasks but have certain knowledge in a broader spectrum of different task types. Both worker types have a low success probability for less preferred categories in common. In addition to the skill set, the worker model stores the measured success rate per category and additional statistics, e.g. the total amount of completed tasks.

By using this model, the worker pool is initialized iteratively. In the first step, a newly created worker is assigned to one of two worker types. The worker type is chosen in respect to the specified share f_{sw} of specialized workers. Accordingly, the amount of *aw* is $1 - f_{sw}$. Based on the type, the preferred categories are selected out of the pool of m categories. The selection follows a random uniform

distribution. In the last step, the success probability for each category is added depending on the favored worker's categories. The iteration of the creation process is completed by adding the worker to the worker pool. These steps are repeated until the predefined number of workers w is reached.

3.3 Recommendation System

In this work we focus solely on illustrating the potential impact of recommendation mechanisms. Thus we decided not to compare current state of the art algorithms but only use a simple content based recommendation algorithm, which recommends tasks based on characteristics of previously completed tasks. The algorithm includes an initialization phase to learn favored task categories of new users. Additionally, we implement a random based task selection as baseline for the evaluation of the recommendation mechanism. The detailed process of each approach is described in the following.

3.3.1 Random selection. The random selection does not consider the qualification of the workers. This means the success rate of each category is not used to determine the workers' best category. The mechanism chooses a task randomly among the available tasks.

3.3.2 Content based selection. The content based algorithm recommends the worker a task of the category in which his success rate (s_c) is greater than a threshold of 50%. We define the threshold at this level, because it is improbable that the worker receives s_c greater than 50% in an unskilled category. In the case that s_c is less than the threshold in all categories, the algorithm computes the category with the highest value of s_c . If there is more than one category with a success rate of the maximal s_c or their value of s_c is greater than 50%, one of them is selected by a random uniform distribution. While choosing tasks, the mechanism considers only category types of which the system contains open tasks. If there are more than one task of the selected category available the algorithm determine the earnings per minute for each campaign and then recommends the best paid task to the worker. We include this aspect, as Schnitzer et al. [14] show that workers are focused on time and money criteria while selecting tasks.

As the algorithm requires a working history, we integrate a training phase for new workers. During this phase the workers have to finish a certain amount of training tasks and their success rate is included in the computation of s_c . Thus, the event based simulation process is extended by an additional step, the training phase. The phase is initiated before starting the worker selection. Here, every worker has to complete the specified amount of training tasks per category. These tasks are not part of the task pool and they only differ concerning the associated category.

3.4 Parameter Settings

As mentioned in the description of the simulation process and its models there are several parameters which can be specified in each simulation run. These parameters are separated into two sets summarized in Table 1. The parameters of the first set define the characteristic of the simulated platform. The amount of categories m describes the diversity of the task types. The share of specialized workers f_{sw} , their success probability p_{sw} and the success probability of the average workers p_{aw} characterize the workers.

Parameter	Role	Description
m	specification	The number of categories in the category pool of a simulation run
f_{sw}	specification	Share of specialized workers
p_{sw}	specification	Success probability of specialized workers in their preferred category
p_{aw}	specification	Success probability of average workers in their favored categories
w	workload	Total amount of workers in the simulation run
t	workload	Mean campaign inter-arrival time in minutes.

Table 1: Functionality of the parameters of the simulation.

The second set of parameters, the total amount of workers w and the mean campaign inter-arrival time specify the workload of the simulated platform.

For our following evaluation we choose the parameters based on the work by Hirth et al. [7]. We use a maximum of 20 categories and realize the varying popularity by adding a higher occurrence to some of these categories. Each category is associated with three campaign types which differ concerning the payment, required time and number of tasks. We choose the payment in a range between \$0.1 and \$1.5 and the required completion time varies from a few minutes up to an hour for an amount of tasks from 30 to 500 per campaign. We use a rate of 0.5 for rejecting unsuitable recommendations by the workers.

3.5 Evaluation Metrics

Since the integration of a task recommendation mechanism may influence the dynamic of the platform, the aim of our analysis is to quantify these influences. Therefore, we define different metrics that consider the viewpoint of the workers. From a worker's perspective his success rate and the earnings are important. To evaluate the influence of the recommendation algorithms on the success rate and the earnings of the workers, we compute the average success rate per hour s of each worker, as well as their average hourly earnings e .

In the following h defines the total simulation time in hours and sn_i is the amount of successfully completed tasks within hour i . Equation 1 describes the computation of s , where n_i represents the number of total completed tasks within hour i . We only consider hours in which the worker completed at least one task.

$$s = \frac{1}{h} \sum_{i=1}^h \frac{sn_i}{n_i} \quad (1)$$

We determine the average earnings per hour e by Equation 2. The payment of task j contained in sn is represented by e_j .

$$e = \frac{1}{h} \sum_{i=1}^h \sum_{j=1}^{sn_i} e_j \quad (2)$$

4 EVALUATION

In this section we evaluate the impact of the task recommendation algorithm in platforms with different characteristics. To identify simulation settings representative for a large number of real-world crowdsourcing platforms, we first analyse the effects of the platform parameters on the workers' success rate and income. Furthermore, we compare the average success rate and the average earnings per

hour of the workers achieved in platforms integrating the recommendation mechanism and the baseline.

4.1 Identification of Key Influence Factors

To evaluate the influences of platform characteristics on the results of the task recommendation mechanisms, we investigate which simulation parameters are the key influences factors. As mentioned earlier, there are two sets of parameters. The first set specifies the platform characteristics, i.e. the amount of categories m , the share of specialized workers f_{sw} , and the success probability of specialized workers p_{sw} and average workers p_{aw} . The second parameter set, describes the workload of the platform. These parameters are the total amount of workers w and the mean inter-arrival time t .

To assess the impact of the different parameters on the success rate s and the earnings e , we run a factor analysis. We define two levels of each simulation parameter and use a 2^k factorial design [10]. This approach requires only a small number of simulation runs to receive results for all setting combinations. For each setting we run 1000 simulations each with a duration of six hours. The transient phase of the simulation is not excluded from the evaluation as it describes the case of new users registering in the system.

Figure 1 shows the influence of the factors on s by using the recommendation approach. Each x-axis of the figure depicts the two levels of the parameter. The y-axis shows the values of s . The results for random based task selection are similar and therefore not shown.

The first graph displays the effect caused by the number m of different task categories. The low level depicts $m = 4$ categories. We choose this value due to the average workers' characteristic of preferring at least two categories. Thus, by using $m = 4$ there are still differences between the average workers concerning the amount of favored categories. The high level $m = 20$ is equal to the maximal amount of defined categories of our simulation model. The value of s observed for $m = 20$ is lower than for $m = 4$. This is due to the availability of tasks in the skilled categories of a worker. The lower the amount of categories the higher the probability that a suitable task is available. In case of four categories the probability of availability of a preferred task of a specialized worker is 25%. The probability in case of an average worker is 50% or more, because he favors between two to four categories.

The second diagram shows the influence of the share of specialized workers f_{sw} . The share of average workers is $1 - f_{sw}$. Thus, the low level of f_{sw} describes a share of 10% of specialized workers and 90% of average workers initialized in the platform. By increasing f_{sw} , a lower success rate s is seen. The difference between the values for the two levels is explained by the main characteristic of specialized workers. They are only skilled in one category. If there is no task available of their preferred category the probability of successfully completing a task in one of the other unskilled categories is very low. Thus, the higher the normalized amount of specialized workers is the lower is the average success rate.

The influence of the success probability of specialized workers p_{sw} is visualized in the third graph. The probability to complete a task successfully is 75% at the lower level of p_{sw} . The upper level specifies a success probability of 90%. As expected there is a higher success rate measured by using the upper level. Here, the specialized worker completes more tasks successfully.

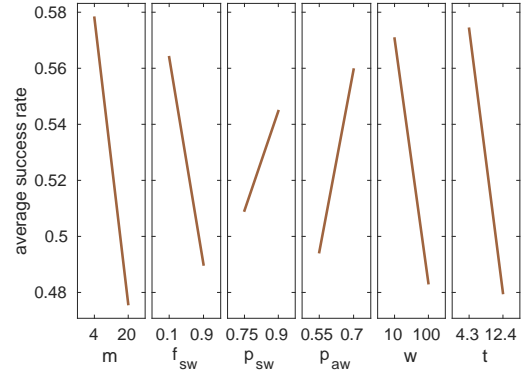


Figure 1: Success rate per hour.

The fourth graph shows the values of s for the two levels of the success probability p_{aw} of average workers. The levels are 55% and 70%. We specify these values to receive a natural order concerning p_{sw} . The upper level affects higher values of s . The reason for this effect is the same as explained in the description of graph three. The higher p_{aw} the more tasks will be completed successfully.

Furthermore, the analysis shows that the total amount of workers w in the platform also affects the average success rate. The effect is shown in graph five. The low level is defined by ten workers and the upper level is represented by hundred workers. These values describe the amount of employees of a small- and mid-sized business. There is a greater value of s observed for the lower amount of workers. This is caused by the workload of the workers. A greater amount of workers decreases the probability that a suitable task is suggested to the requesting worker.

A similar effect is seen for the different levels of the mean inter-arrival time of campaigns t , which is displayed in graph six. As mentioned the inter-arrival time is described by a negative exponential function. Thus, the factor levels vary regarding the mean t of this function. The upper level of t is about 12.4 minutes. It is based on the results of the analysis of the campaign inter-arrival time of Microworkers. The lower level describes an average inter-arrival time of 4.3 minutes which is approximately one third of the upper level. The value of the average success rate is greater for the shorter inter-arrival time than for the upper factor level. This is due to the amount of open tasks in the platform. The lower the inter-arrival time the more campaigns will be created and the more tasks will be available in the platform. Thus, the probability of selecting tasks which fit the skills of the requesting worker is very high.

Concluding the average success rate is influenced positively by a small amount of categories, a small share of specialized workers, a high success probability of specialized and average workers, and a small total amount of workers, as well as a short campaign inter-arrival time.

The results of the factor analysis concerning the average hourly earnings of the workers are similar to the influences as described for the average success rate per hour. The similarity is caused by the dependency between the successful completion of tasks and getting paid. This means by completing more tasks successfully the

Platform Type	m	f_{sw}	p_{sw}	p_{aw}	w	t
Specialized platform	4	0.1	90%	70%	10	4.3
Unspecialized platform	20	0.9	75%	55%	100	12.4

Table 2: Settings of a specialized and an unspecialized platform, defined by the amount of categories m , the share of specialized workers f_{sw} , the success probability of specialized workers p_{sw} and average workers p_{aw} , the amount of workers w , and the mean inter-arrival time t .

earnings increase. Thus, each factor which influences the success rate positively will also affect the earnings in a positive way.

4.2 Deductive Key Scenarios

To evaluate the influence of recommendation algorithms in platforms with different characteristics and different workload we combine the levels of the parameters which affect the success rate and earnings positively and the levels which influences are negative. The resulting simulation settings are shown in Table 2. Having a closer look at the resulting platform characteristics, we can identify two platform types.

The first platform type is specialized on a small amount of different categories and the amount of registered workers is low. Due to the small amount of categories they are not specialized on one category. This means the share of average workers is great. In addition, they are very high skilled in their preferred categories. Consequently, the probability of completing tasks of favored categories successfully is very high. The inter-arrival time of campaigns is low. The small amount of workers and the large amount of campaigns defined by the short inter-arrival time describes a high workload of the platform. This workload influences the success rate and the earnings positively.

The other platform type described by the second setting combination shown in Table 2, represents a non-specialized crowdsourcing platform. The platform offers a great amount of various task categories, which results in a lower success rate and hourly earnings. This results in a specialization of a great part of workers specified by $f_{sw} = 0.9$. Overall the success probability of all workers is lower than in the other platform type. However, there are more workers registered in the platform. Due to the longer inter-arrival time, there are less campaigns created in this platform type and thus, the workload is low.

The workload of both platform types can be varied by changing the ratio of workers and created campaigns. This means, by the reduction of registered workers and the decrease of the mean inter-arrival time, the workload increases.

In the next subsection we investigate the impact of the task recommendation algorithm and the baseline on the average success rate per hour and the hourly earnings per worker by setting up the simulation model with the parameters of the two platform types.

4.3 Influence on Success Rate and Earnings

To evaluate the impact of the recommendation system on the average earnings e of each worker in combination with the received average success rates s , we normalize the hourly earnings by the highest seen income per simulation run. The maximal amount of

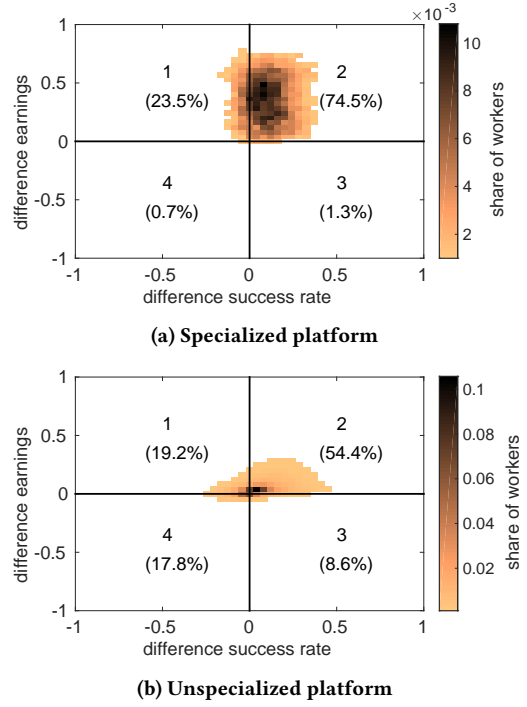


Figure 2: Differences between random and content based recommendation concerning the success rate and the earnings

money to earn per run depends on the task categories which are contained in this model. Based on these values we compute the differences of e and s gathered while using the content based system and the random based approach. The differences quantify the improvement when using the content based mechanism. Larger values of these differences imply a greater enhancement. This means the hourly wages and the success rate is higher.

To obtain comparable results we run both task selection mechanisms on the same generated models. This includes all model configurations which means tasks and workers.

The improvement per worker measured in a specialized platform is visualized in a 3D-histogram in Figure 2a. The colored areas describe the amount of workers with a specific difference of s and e normalized by the total amount of workers. The darker the color of an area, the greater is the share of workers. We omit outliers which are represented by areas containing a share of workers less than 1%. By separating the figure in four sections, we group the workers based on their difference values. Thus, we can analyse the amount of workers with an increase of s and e , an increase of only one of these values or those who are earned less in combination with a lower success rate.

We observe a small negative average difference of the success rate of workers of section 1 in the upper left. Thus, the earnings are only increased. The workers of the second section which means the upper right quadrant, benefit concerning their success rate and their earnings when using the recommendation system. Here, the share of workers is 74.47%. The average success rate of the workers in the lower right section (3) is increased whereas their earnings are

not significantly decreased. There is no improvement for workers residing in section 4 in the lower left. The share of workers of section 3 and 4 is negligible small.

Concluding, the usage of the content based system increases the average earnings and the average success rate of 74.47% of the workers during a simulation time period of 6 hours in a specialized platform. For 23.55% of the workers only the earnings are increased while their success rate is not significantly decreased.

Figure 2b shows a 3D-histogram of the workers registered in an unspecialized platform. In this case the upper left quadrant (1) contains 19.19% of the workers. The second section in the upper right which describes the case that s and e are increased contains 54.4% of the workers. 8.58% of the workers are grouped in section 3. The worst case is shown in the lower left section (4). Here, the earnings and the success rate are slightly decreased for 17.83% of the workers. In conclusion the content based system achieves an increase of the earnings and the success rate for 54.4% of the workers. The increase of the success rate is higher than for the earnings, due to the amount of available tasks in the platform specified by the workload. The probability that tasks of different campaigns of favored categories are available, is very low. Thus, the recommendation mechanism suggests the tasks without considering their payment.

Concluding, we observe that e and s are affected by integrating different task recommendation algorithms in both platform types. The content based technique results in a higher success rate and income for more workers than the baseline. The analysis of the influence on the hourly earnings e shows also an increase of the earnings for the content based system by comparing the values to the random approach.

5 CONCLUSION

Recommendation systems are nowadays integrated in many services and applications to help coping with the tremendous amount of data and items available. This makes them also likely to be valuable tool in commercial crowdsourcing platforms, to help mapping tasks to workers who have the skills to complete them successfully. Even if there already exist several work in this direction, no systematic evaluation was available on how those systems affect workers on the platform.

To tackle this question, we built a simulation model of a crowdsourcing platform including recommendation mechanisms. Based on the analysis of influences of the simulation parameters, we identified key scenarios which describe two different platform types. We investigated the impact of a content based recommendation algorithm concerning the workers' success rate and the earnings.

The analysis of the results shows that hourly earnings and success rates are impacted by recommendation in both scenarios. For the non specialized platform scenario, the success rates and earnings are positively affected for a significant amount of workers, while a small share of workers (17.83%) is negatively affected.

There are still several quality criteria and aspects which could be investigated by using the simulation model. One aspect is the fairness of the task distribution between the workers. Furthermore, the variety of recommended tasks to workers who are skilled in more than one category could be evaluated.

ACKNOWLEDGEMENT

This work is supported by Deutsche Forschungsgemeinschaft (DFG) under Grants TR 257/38-2. The authors alone are responsible for the content.

REFERENCES

- [1] Vamshi Ambati, Stephan Vogel, and Jaime G. Carbonell. 2011. Towards Task Recommendation in Micro-Task Markets. In *Workshop on Human Computation*.
- [2] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated experiments on Ad privacy settings. *Privacy Enhancing Technologies* (2015).
- [3] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudre-Mauroux. 2013. Pick-a-crowd: tell me what you like, and i'll tell you what to do.. In *International World Wide Web Conferences*.
- [4] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *International Conference on Knowledge Discovery and Data Mining*.
- [5] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. 2014. A taxonomy of micro-tasks on the web. In *Conference on Hypertext and social media*.
- [6] David Geiger and Martin Schader. 2014. Personalized task recommendation in crowdsourcing information systems - Current state of the art. *Decision Support Systems* 65 (2014).
- [7] Matthias Hirth, Tobias Hofffeld, and Phuoc Tran-Gia. 2011. Anatomy of a crowdsourcing platform-using the example of microworkers.com. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*.
- [8] Chien-Ju Ho and Jennifer Wortman Vaughan. 2012. Online Task Assignment in Crowdsourcing Markets.. In *International Conference on Artificial Intelligence*.
- [9] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. *Machine Learning and Knowledge Discovery in Databases* (2012).
- [10] Averill M. Law and David M. Kelton. 1999. *Simulation Modeling and Analysis* (3rd ed.). McGraw-Hill Higher Education.
- [11] Habibur Rahman, Lucas Joppa, and Senjuti Basu Roy. 2016. Feature Based Task Recommendation in Crowdsourcing with Implicit Observations. *arXiv preprint arXiv:1602.03291* (2016).
- [12] Andrea Romei and Salvatore Ruggieri. 2014. A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review* 29, 05 (2014).
- [13] Senjuti Basu Roy, Ioanna Lykourantzou, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. 2015. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal* 24, 4 (2015).
- [14] Steffen Schnitzer, Christoph Rensing, Sebastian Schmidt, Kathrin Borchert, Matthias Hirth, and Phuoc Tran-Gia. 2015. Demands on task recommendation in crowdsourcing platforms-the worker's perspective. In *CrowdRec Workshop*.
- [15] Christian Schwartz, Kathrin Borchert, Matthias Hirth, and Phuoc Tran-Gia. 2015. Modeling crowdsourcing platforms to enable workforce dimensioning. In *International Conference on Telecommunication Networks and Applications Conference*.
- [16] Jie Yang, Judith Redi, Gianluca DeMartini, and Alessandro Bozzon. 2016. Modeling Task Complexity in Crowdsourcing. In *Conference on Human Computation and Crowdsourcing*.
- [17] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2011. Task matching in crowdsourcing. In *International Conference on Cyber, Physical and Social Computing*.
- [18] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2015. Taskrec: A task recommendation framework in crowdsourcing systems. *Neural Processing Letters* 41, 2 (2015).
- [19] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2016. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. *arXiv preprint arXiv:1610.08452* (2016).

The Price of Fairness in Location Based Advertising

Chris Riederer
Columbia University
New York, NY 10027
mani@cs.columbia.edu

Augustin Chaintreau
Columbia University
New York, NY 10027
augustin@cs.columbia.edu

ABSTRACT

Firms use massive amounts of personal data to decide which advertisements to show to an individual, raising concerns of fairness and algorithmic bias. Previous work has proposed techniques to make machine learning more fair through awareness of the protected attributes of user data. However, these studies have either focused on specific tasks, been primarily theoretical, or have ignored the highly important domain of location-based advertising.

In this work, we present an empirical analysis of the impact of fairness on advertising revenue using a real world example: location based ad personalization for users of Instagram. We empirically analyze the potential for inadvertent discrimination among gender and race in location-based systems, additionally showing the impact of location representation on fairness. Furthermore, we apply fairness techniques to analyze how revenue is affected when both individual and group fairness guarantees must hold. Though this work is a grounding for research into fairness in location-based ads, our methodology applies to more general advertising tasks.

ACM Reference format:

Chris Riederer and Augustin Chaintreau. 2017. The Price of Fairness in Location Based Advertising. In *Proceedings of ACM RecSys conference, Como, Italy, August 2017 (FATREC'17)*, 5 pages. <https://doi.org/10.18122/B2MD8C>

1 INTRODUCTION

Every day, personal data becomes more broadly available and its use in analytics and advertising clearly generates large sums of wealth. What is perhaps less clear is how tools to prevent discrimination against vulnerable populations can keep up with the growth in algorithmic decision-making based on this personal data. Here we focus on informing what can *practically* be done to guarantee fairness when location data is used in targeted advertising. We choose this application for multiple reasons: It is increasingly common as location-based personalization reaches a large part of the population and it is hard to evade. As we empirically demonstrate, mobility data has great benefits but raises many concerns in the way it is currently used. Perhaps more importantly, we show that many of the hardest challenges previously addressed in theoretical terms can be quantified in this scenario. For instance, this brings us to revisit questions like “What constitutes a practical definition of fairness?”, “What should we know or trust about those exploiting the data?”, “What is the gain we lose when some definition of fairness must be enforced?”

Let us describe a motivating example where disparate outcomes in targeted advertising is undesirable. For instance, consider a website advertising hiring opportunities to users; its goal is to optimize for relevance as long as disparate outcomes among genders and races are avoided. Why would such a system pose new challenges? First, previously proposed solutions focus on reconciling learning and fairness for *specific tasks for a single party* [1, 2, 10, 11]. For instance, how to increase loan repayment while satisfying equality of treatment or opportunity. In contrast, data providers interact with myriad third parties each leveraging data for different learning tasks. Second, as is commonly the case for online data providers, data about individuals are sparse and naturally represented in high dimensions. This contrasts with solutions designed to learn from a few structured features available for all users, such as exam scores. Additionally, leveraging data at large scale invariably means that computational complexity becomes a severe constraint, so each optimization to reconcile fairness with accuracy will rely on efficient approximation.

These challenges, however, do not imply that no solutions can be found to deploy fair targeting. The direction we examine here is to transform location data before they are used to train and target individuals. If the transformation and targeting satisfies some conditions (see background below), then fairness can be guaranteed for *any* task. As we demonstrate, much of the gains from targeting is preserved. For concreteness and simplicity, we focus in this short article on the simplest transform where details of mobile data are remove by grouping records into larger location cells.

2 BACKGROUND

In our work, we use the definitions of “Fairness Through Awareness” [3], distinguishing between fairness at an individual level and at a group level, which we describe in detail below.

Individual fairness. The main principle is that similar people should see similar outcomes. More rigorously, we consider a classification setting where individuals (denoted by the set V) are mapped to probability distributions over outcomes A . For simplicity, throughout this work we will say each outcome is the decision of whether to show either a generic or targeted ad, and denote these outcomes as $A = \{0, 1\}$ with $A = 1$ corresponding to the decision to show a targeted ad and $A = 0$ a generic ad instead. The space of probability distributions defined on A is $\Delta(A)$. From our point of view, a machine learning algorithm using data from the mobile ad-network defines a mapping $M : V \rightarrow \Delta(A)$. A difference score between individuals is denoted by $d : V \times V \rightarrow [0; 1]$ and a difference score between probability distributions is D . Throughout this paper, without loss of generality, as a choice to measure the distance between probabilistic outcomes we will use D_{TV} , the distance of total variation (equivalent to one half the \mathbb{L}_1 norm) though others can be used. It is defined as: $D_{TV}(P, Q) = \frac{1}{2} \sum_{a \in A} |P(a) - Q(a)|$.

Given these definitions, an algorithm is *individually fair* if for all individuals x and y , we have

$$D_{TV}(M(x), M(y)) \leq d(x, y) \quad (1)$$

Intuitively, this says that an advertising system must show similar sets of ads to similar users, and mathematically, this means that that an algorithm mapping users to distributions over outcomes must be Lipschitz continuous.

[3] shows that it is possible in polynomial time to find a mapping M that is both individually fair and maximizes a linear objective function (such as expected revenue) using a linear program.

Group fairness. In contrast to individual fairness, [3] defines two groups of users S and T as having statistical parity up to bias ϵ when:

$$D_{TV}(E_S[M], E_T[M]) \leq \epsilon \quad (2)$$

where E_S and E_T denotes the expectation of ads seen by an individual chosen uniformly among S and T . This definition implies that the difference in probability between two groups of seeing a particular ad will be bounded by ϵ . Note that individual fairness does not imply group fairness, and vice versa. A natural question is: "When can both individual fairness and statistical parity be achieved simultaneously"? To guide the design of a mobile platform one can use the following result introducing $d_{EM}(S, T)$, the Earth Mover's Distance [6] between S and T .

THEOREM 1. *Given a distance $d \leq 1$, among all algorithms M that are individual fair, for any subsets of users S, T we have*

- (i) *It always holds that $D_{TV}(E_S[M], E_T[M]) \leq d_{EM}(S, T)$,*
- (ii) *there exists M such that $D_{TV}(E_S[M], E_T[M]) = d_{EM}(S, T)$.*

3 DATA DESCRIPTION

To understand the important trade-offs facing advertising platforms, we collected a behavioral dataset linked to race and gender information. We obtained publicly available data from Instagram, a popular image sharing social network. Instagram data includes behavioral data such as locations and short texts of describing activities as well as the photos themselves which provide information through the use of computational vision techniques.

3.1 Methodology

We gathered metadata (such as time of photo, URL of image, tags, location, etc.) for all photographs of a "root" user, Kevin Systrom, the founder of Instagram. We then randomly sampled user profiles from those who had commented or liked his photos and gathered their metadata. We repeated this process, randomly sampling user IDs of those commenting or liking photos of any crawled profiles, obtaining the metadata of 115,796,284 for 260,389 different profiles. Systrom is a popular Instagram presence (1.4M followers) and a wide variety of users comment on his photos, seemingly to communicate with the platform, making him a good starting point for a random crawl. No images were downloaded from Instagram.

Location. Of our 115 million photo information dataset, 16,537,404 were geotagged for 162,549 users. In order to study advertising that micro-targets small granularity locations, we narrowed our focus to two major United States cities, New York City and Los Angeles, a typical practice. Using only photos located in the bounding boxes of those two cities, we created two subsets: New York had 22,300

Dataset	Number Users	Number Checkins	Labeled Gender	Labeled Race
New York	22,300	707,265	10,388	902
Los Angeles	20,724	776,065	9,748	851

Table 1: Overview of dataset used in study.

users with 707,265 photos and Los Angeles had 20,724 users with 776,065 photos.

Tags. Like other social networks, Instagram users label their content with "hashtags", which label topics for the photo, make photos more easily searchable, or let the user express him- or herself. As we discuss in a later section, we use these tags later as part of our location-based advertising model.

3.2 Labeling

Labeling gender. To label our the gender of the users in our dataset, we applied the methodology of Mislove et al. [4]. We obtained the number of babies born by name, gender, and year of birth in the United States via Social Security data¹, assigning a gender to users with a first name for which there were both at least 50 births and 95% of recorded births were one gender. Out of our entire dataset of 260 thousand users, this labeled 92,935 profiles (35%). In our New York City subset, 10,388 were labeled with gender, 5,471 female and 4,917 male. In Los Angeles, 9,748 users labeled with gender: 4,965 female and 4,783 male.

Race labeling. We labeled the race of profiles based on face recognition software, similar to prior work [5]. The Face++ API (www.faceplusplus.com) recognizes faces in images, additionally providing demographic information, labeling the race of users from one among Asian, Black, and Caucasian. Although we did not download any photos, our metadata included publicly accessible URLs of images, which we could pass to the Face++ API. We ran this software on the first 500 photographs of a subset of our New York and Los Angeles users, labeling a profile with a binary race classification (Caucasian or minority) that appeared most frequently in their photographs. This labeled 902 users in our New York dataset; 746 labeled Caucasian and 156 from minorities, and 851 users in Los Angeles; 710 Caucasian and 141 minority.

Evaluation with manual labeling. To provide ground truth validation of our more scaled labeling techniques, two research assistants labeled a randomly selected subset of 200 profiles for gender and race. After filtering for private, deleted, or business profiles, 194 profiles remained. Of our 194 human-labeled profiles, 86 users had first names recognized by our methodology. Of these, 84 out of 86 (97%) agreed, giving us high confidence in the precision of our gender labeling approach. For race labeling, our computational vision approached agreed with human labelers 89.7% of the time, comparable to other works that report that Face++ has high levels of accuracy for race labeling.

4 MOBILE ADVERTISING MODEL

In order to analyze the trade off between fairness and revenue, we model a location-based advertising system using our dataset. We focus on this domain due to its importance (38% of all smartphone advertising used location targeting in 2016), and its potential for

¹ Available at <https://www.ssa.gov/oact/babynames/limits.html>

discrimination as location is highly sensitive and often correlates with sensitive traits such as race or income [8]. We simulate a system with the following problem: Given a user's locations from previous check-ins, predict what topics a user will be interested in. Such a prediction could allow a service to better target ads.

4.1 User and Location Representation

We represent individuals in terms of their visits to different locations. We map locations to an index j . Each user is represented as an array, with index j set to 1 if the user has checked in at location j and a 0 otherwise. In our original dataset, locations for each photo are latitude-longitude pairs, and here we discretize these by truncating these coordinates to a certain level of precision. In different analyses we vary this precision to study how fairness and revenue is impacted by granularity of location representation. Using fewer digits implies a lower granularity, which is better for privacy but less specific and hence likely less useful for advertisers. We vary the cell sizes from 0 decimal places (e.g. (-74., 40.) is a cell; cells have sides of length roughly 111km) to 4 places (e.g. (-73.9989, 40.7245) is a cell; cells have sides of roughly 10m). We additionally conducted our analysis representing users with a histogram of frequencies of visits to each location as opposed to binary representations, but the results were similar and we omit them due to space.

4.2 Interest Prediction

After defining how users are represented, we use these feature to predict if a user is interested in several topics, utilizing Instagram's hashtags for ground truth. Hashtags, used on several platforms such as Instagram and Twitter, are ways for users to associate topics with their post. Examples include a user tagging a picture of food with "#food" or of himself with "#selfie". We use three different tags: #fashion, #travel, and #health.

We trained a model predicting a user's likelihood to post each of the three tags using a user's location visits as features and whether or not they had used a tag as labels. To avoid overfitting we regularized each model using ridge regression (i.e. \mathbb{L}_2 penalty) and conducting three way cross validation, picking the parameter that maximized performance on the training set. All training was conducted using the scikit-learn python package.

4.3 Performance and Revenue Estimation

We evaluate our models in two ways: in traditional machine learning terms and for their ability to improve revenue in an advertising simulation. We use AUC as a metric to understand our classifier performance due to its standard acceptance and our class distributions being highly skewed. For all three tags and both cities, AUC is 0.5 at the broadest granularity, meaning our model is no better than random guessing. However, as the number of digits increases, so does AUC. In NYC, our classifiers have AUCs of 0.82, 0.92, and 0.65 for fashion, health, and travel, respectively, and in LA, we report AUCs of 0.83, 0.92, and 0.68.

Moving beyond classifier performance, we estimated the impact of granularity on revenue. Earlier, we distinguished between generic and targeted advertisements. Based on estimates generated from the Facebook ad tool², we said that the cost per click (advertiser

revenue) for a targeted ad was \$2 and the revenue for a generic ad was \$1. In our model, a generic ad always generates revenue, and a targeted ad only generates revenue if the user is indeed interested in a topic, and so the system will only show a targeted ad to a user if the expected revenue justifies the risk of receiving no revenue. Using this model, a predictor using the finest granularity of 4 digits generated \$1021, \$994, and \$906 in revenue for fashion, travel, and health, respectively, over a baseline of displaying generic display \$902. The results were similar for LA.

5 EVALUATION

5.1 Balancing Fairness and Revenue

We now consider revenue maximization under the constraint of individual fairness. In Sec. 2 we referenced how this could be achieved after the choice of a distance function between outcomes, a distance function between users, and a linear objective function. Our choice of D , the distance between distributions of ads, is $D_{TV}(P, Q) = \frac{1}{2} \sum_{a \in A} |P(a) - Q(a)|$. For our choice of d , the distance score between users, we again use the distance of total variation, this time upon the histogram of visits to locations between each pair of users using the representation of users defined in Sec. 4.1. Our objective function is to maximize expected revenue, as defined as $\sum_{x \in V} g \cdot \mu_x(0) + t \cdot \mu_x(1)$ with g , the revenue of a generic ad, set to 1 and t , the revenue of a targeted ad set, to 2. After these choices, the linear program chooses a probability of showing a targeted ad to a user to maximize revenue under the constraints of similar users seeing similar ads.

In order to make the trade-off between revenue and fairness more fluid, we differ from prior work and introduce a new parameter k into Eq. 1:

$$D_{TV}(M(x), M(y)) \leq k \cdot d(x, y) \quad (3)$$

A large k means more flexibility in ad assignment but less individual fairness; $k = \infty$ means identical users can see completely different ads. In contrast, a low value of k constrains the problem more, with $k = 0$ meaning all users must have the same ad distribution.

We run this linear program for both cities at all granularity levels and for multiple choices of k . We then compute a real revenue with the function $\sum_{x \in V} g \cdot \mu_x(0) + t \cdot \mu_x(1) \cdot \mathbf{1}_{x \in I}$ with the set I denoting users who actually posted the target tag. Due to the number of constraints growing quadratically with the number of users, Here we are only able to present results for fairness by race and leave detailed analysis of gender for later work.

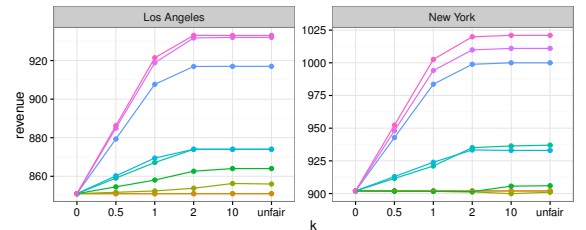


Figure 1: The impact of k and granularity impact on revenue.

Fig. 1 displays the impact of k and granularity on revenue for both cities with the tag fashion. The x axis corresponds to the choice of k used in the linear program. The y axis represents the actual

²<https://www.facebook.com/business>

revenue of the ad assignments output by the LP. Color denotes the granularity of location. The graph demonstrates again how finer granularity can increase revenue. In both NYC and LA, at nearly all values of k , a higher granularity corresponds to higher revenue. Another important takeaway is the shape of the lines. The revenue at $k = 2$ is nearly identical to the revenue at all higher amounts of k . The revenue declines rapidly at $k = 0$, where all individuals have the same distribution, and $k = 0.5$. The increase in revenue from $k = 1$ to higher values of k is significant but not a large portion of the highest optimal revenue, suggesting a good potential value due to its balance and simplicity.

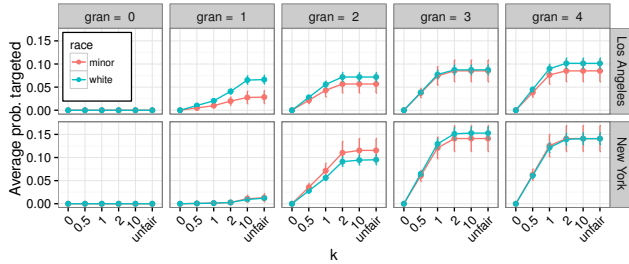


Figure 2: The impact of k and granularity on fairness.

We next examine the impact of k and granularity upon fairness. In Fig. 1, the x axis again corresponds to value of k . Color corresponds to race, with blue associated with caucasians and red associated with minorities. The y axis now corresponds to the average probability that users of the class saw a targeted ad, with error bars corresponding to standard error of the mean. Each facet represents a different level of granularity.

At lower levels of granularity, all users have similar low-resolution representations and thus it is difficult for our click predictor and then LP to risk displaying targeted ads, instead showing generic ads at all values of k . At medium level granularities, we see the algorithm begin to assign the ad to a small number of users and additionally the lines for each class to diverge, signally a rising level of group unfairness. Interestingly, in both graphs, the lines converge to be near identical at finer levels of granularity, at 4 digits for NYC and 3 and 3.5 digits for LA. This could be caused by mid-range granularities being associated more with neighborhoods, whereas very fine granularities will correspond to more exact venues, removing rougher associations of neighborhoods around areas with certain tags and narrowing them down to more specific places (e.g. 2 lat-long digits corresponds to roughly 1km, 4 to 10m).

5.2 Bounding Fairness

For two demographic attributes, race and gender, we compute the Earth Mover’s Distance, using the pyemd package [6, 7]. More precisely, for race we calculate the EMD between two probability distributions, one over Caucasian users and the other over Non-Caucasian users, with the “distance” between users defined as the distance of total variation of the histogram of their location visits. Similarly, for gender we calculate the EMD between the distribution of female and male users. As mentioned in Section 2 we represented locations as “cells”, assigning a photograph to a cell by truncating the latitude-longitude coordinates by a varying amount.

The large number of users labeled with gender presented a difficulty for our EMD calculation as Earth Mover’s Distance does not scale well. We use agglomerative clustering [9] to approximate EMD. We found this technique that groups individuals into “points” is well suited to our problem due to nonuniform cluster sizes.

We add a mechanism to cope with statistical parity, as it may create a spurious statistical bias between finite size groups, even when the expectations among those groups are equal. In addition to computing EMD between demographic groups, we also computed EMD between randomly created groups with the same size as our demographic groups.

In Fig. 3 we show the result of this process. The x -axis shows the granularity in terms of latitude longitude decimal places. The y axis shows the EMD. Lines are colored according to demographic, and a dashed line indicates random grouping of users as opposed to grouping by demographics. To put the EMD numbers into perspective, on the lower end, an EMD of 0.05 means one group may be seeing a targeted ad 5% more often. At the higher end of 0.8, users across the two groups are seeing quite different sets of ads.

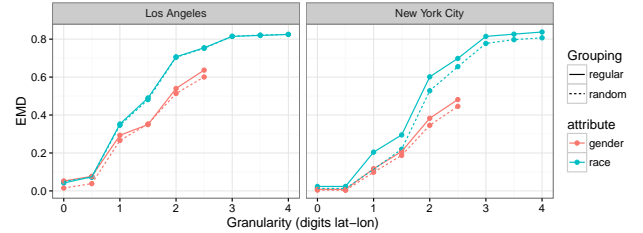


Figure 3: Risk vs. Granularity

In New York for race, the random line is clearly below the regular line, providing some evidence of real differences between the demographic groups as opposed to an artifact of sparsity. The line for gender is additionally more separate than its counter-part in Los Angeles. This is possibly due to the much higher density in New York. As all users begin to have high difference scores from one another, caused by having no overlapping locations due to low density, all label assignments will be indistinguishable from each other. Gender overall seems to show a weaker separation between the real EMD and the random EMD.

The EMD increases as the data becomes more precise. One limitation of this study is that the distance d we chose does not distinguish two users who have nearby but non-intersecting visits and users who are on the opposite side of the city. Different choices of d with true geographical distance may refine those results.

6 CONCLUSION

In this work, we showed the impact of granularity on ad targeting, demonstrated the impact of fairness algorithms on a real world behavioral dataset, and explored a utility-fairness trade-off. There are many possible future directions. All results should be reproduced on larger datasets and different classes. One idea is to reformulate the problem in terms of *where* ads are shown or how users are reached, as opposed to focusing on the individuals. Building on our results, we also hope to create scalable algorithms for debiasing representations of users that work with sparse, large behavioral datasets.

REFERENCES

- [1] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *Advances in Neural Information Processing Systems (NIPS)*, July 2016.
- [2] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Proceedings of Workshop FATML*, stat.AP, October 2016.
- [3] Cynthia Dwork, M Hardt, T Pitassi, and O Reingold. Fairness through awareness. In *ITCS '12 Proceedings of the 3rd conference on Innovations in Theoretical Computer Science*, 2012.
- [4] Alan Mislove, S Lehmann, Y Y Ahn, and J-P Onnela. Understanding the Demographics of Twitter Users. *ICWSM*, 2011.
- [5] S Nilizadeh, A Groggel, P Lista, and S Das. Twitter's Glass Ceiling: The Effect of Perceived Gender on Online Visibility. *Proceedings of the International Conference Weblogs and Social Media (ICWSM)*, 2016.
- [6] O Pele and M Werman. Fast and robust earth mover's distances. *2009 IEEE 12th International Conference on Computer Vision*, 2009.
- [7] Ofir Pele and Michael Werman. A linear time histogram metric for improved SIFT matching. In *Proceeding ECCV '08 Proceedings of the 10th European Conference on Computer Vision*, pages 495–508. Hebrew University of Jerusalem, Jerusalem, Israel, December 2008.
- [8] Christopher J Riederer, Sebastian Zimmeck, Coralie Phanord, Augustin Chaintreau, and Steven M Bellovin. I don't have a photograph, but you can have my footprints.: Revealing the demographics of location data. In *Proceedings of the 2015 ACM on Conference on Online Social Networks*, pages 185–195. ACM, 2015.
- [9] Joe H Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244, March 1963.
- [10] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna Gummadi. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In *WWW '17 Proceedings of the 26th International Conference on World Wide Web*, October 2016.
- [11] Jiaming Zeng, Berk Ustun, and Cynthia Rudin. Interpretable Classification Models for Recidivism Prediction. *FATML*, March 2015.

Fair Sharing for Sharing Economy Platforms

Abhijnan Chakraborty

Max Plank Institute for Software Systems, Germany
Indian Institute of Technology Kharagpur, India

Aniko Hannak

Central European University, Hungary

Asia J. Biega

Max Planck Institute for Informatics
Saarland Informatics Campus, Germany

Krishna P. Gummadi

Max Plank Institute for Software Systems, Germany

ABSTRACT

Sharing economy platforms, such as Airbnb, Uber or eBay, are an increasingly common way for people to provide their services to earn a living. Yet, the focus in these platforms is either on the satisfaction of the customers of the service, or on boosting successful business transactions. However, recent studies provide a multitude of reasons to worry about the providers in the sharing economy ecosystems. The concerns range from bad working conditions and worker manipulation to discrimination against minorities. This is worsened by the fact that the algorithms used for matching customers and providers, that de facto decide the amount of exposure each provider receives, are proprietary and non-transparent.

In this position paper, we propose a novel framework to think about fairness in the matching mechanisms of online sharing economy platforms. Specifically, we focus on various fairness guarantees from the providers' perspective. Our notion of fairness relies on the idea that, spread over time, all providers should receive the amount of exposure proportional to their relevance or the utility they provide. We postulate that by not requiring every match to be fair, but rather distributing the fairness over time, we can (i) give better guarantees in terms of the overall benefit for the providers and the customers, (ii) make use of implementations from a long line of research concerned with fair division of constrained resources. Overall, our work takes the first step towards rethinking fairness in online sharing economy systems with an additional emphasis on the well-being of providers, and provides insights into parallels with well-established practical implementations in other domains.

ACM Reference format:

Abhijnan Chakraborty, Asia J. Biega, Aniko Hannak, and Krishna P. Gummadi. 2017. Fair Sharing for Sharing Economy Platforms. In *Proceedings of FATREC Workshop on Responsible Recommendation at RecSys 2017, Como, Italy, August 2017 (FATREC 2017)*, 4 pages. <https://doi.org/10.18122/B2BX2S>

1 INTRODUCTION

While the *sharing economy* or *two-sided market* has traditionally been thought of as a movement towards more democratized marketplaces, increasing number of studies and articles are concerned with the potential discriminatory effects of some sharing economy giants such as Uber or AirBnB [8, 21]. Sharing economy is loosely defined as *peer-to-peer-based activity of obtaining, giving, or sharing the access to goods and services*, which is coordinated through a

web-based platform or a mobile-app [12, 28]. The rise of social technological systems and online platforms has enabled it to become a major competitor to the traditional (B2C) economic model. According to a report by the United States Department of Commerce from 2016, the spending in the most common areas of sharing economy was 5% of the total economy, and this is predicted to grow to 50% by 2025 [23]. Recognizing the opportunity, startups that build sharing economy platforms have mushroomed and become a mega trend among Silicon Valley investors, attracting millions of dollars in venture capital funds [26].

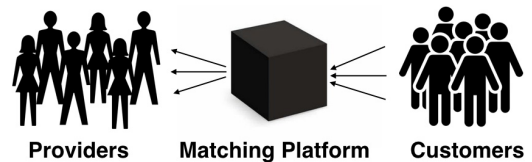


Figure 1: Different stakeholders in sharing economy framework: providers, customers and the platform.

In the sharing economy framework, there are three stakeholders: (i) **providers** of goods and services, (ii) **customers** who pay for them, and (iii) the **platform** which provides the *matching* between the providers and customers. As depicted in Figure 1, the platform lies in the center of the framework, enabling the providers and the customers to connect and do business. Crucially, the platform has a control over the exposure of service providers to potential customers and vice-versa. For example, Uber matches drivers with passengers “under-the-hood” [8]. In case of Airbnb or different freelance websites, customers have more control over the choice of the provider, but the platform still decides how much exposure and attention each provider gets and which customers they are shown to. Hence, a pertinent question to ask in this context is *what should be the objectives of the matching platform?*

Most matching platforms today try to maximize the *utility* for the customers, driven by the *customer is always right* philosophy [16]. The underlying idea is that the party paying the platform for using the service (passengers in Uber, renters in Airbnb, employers in Freelancer, or buyers on Amazon) should be most satisfied, yielding the most revenue for the platform. However, the providers may get squeezed in the process. For example, Uber drivers complain that they have to work long hours for little pay [11]. Similarly, sellers in ecommerce marketplaces like Amazon complain about increasing participation costs and declining profits [25].

Moreover, recent studies have shown that inequalities from the offline world easily transfer to these online platforms, making it harder for minorities to succeed. This is, for instance, partly due to

biased hiring choices of the employers (job platform customers) [1, 10, 14, 24], or biased evaluations and ratings of providers, based on their gender, race, etc. [13, 15]. However, even more worrying are the effects of the big data algorithms that the platforms use to match customers with providers. A lot of prior works have raised concerns regarding the biases algorithmic decisions carry [5–7, 9, 18, 22]. Likewise, the algorithms deployed by the matching platforms may reproduce or even reinforce biases that are present among the customers, leading to a *rich-get-richer* effect for the providers. Thus, while optimizing for the most profitable matching, unfortunately the minorities among the providers may not receive their deserved attention.

In this position paper, we argue for establishing the notion of *fairness of matching mechanisms in sharing economy platforms*. The related research questions can be threefold:

RQ1. What notions, measures and criteria of fairness should be applied on these platforms?

RQ2. Do the existing sharing economy platforms satisfy them?

RQ3. How can we (re)design platforms to satisfy the fairness criteria identified in RQ1?

In this work, we focus on answering RQ1, and leave the other research questions for future work. Fairness of the matching can be ensured by optimizing for: (a) fairness for providers, (b) fairness for customers, or (c) fairness for both customers and providers. In this work, we focus on goal (a), investigating the trade-off between utility for customers and fairness to providers, and leaving other scenarios as subjects of future work.

In a marked departure from past efforts on defining fairness in search and recommendation systems [17, 19], we add a **temporal dimension to the notion of fairness**. We argue that a *fair matching platform* would be something that distributes the exposure of providers to customers *over time in a fair way*. The exact fairness notions are discussed in Section 2.

The advantage of introducing the time dimension into the notion of fairness in sharing economy platforms is two fold:

- (i) The fairness notion becomes more relaxed than the constrained requirement of being fair with respect to every matching, and
- (ii) Abstractly, the problem maps to one of fair division of a constrained resource (in this context, the exposure of providers to customers) over time. There is a long line of work on bringing fairness in generalized processor sharing algorithms [2, 29], which could be applied in a temporally fair matching system.

In summary, in this position paper, we make two contributions: (i) we provide a systematic way to think about fairness in sharing economy platforms, and (ii) by introducing the temporal dimension to the notion of fairness in such platforms, we enable reuse of the existing techniques for fairness in processor scheduling. In future work, we would go beyond fairness for providers, and consider fairness for all members of the sharing ecosystem. Subsequently, we would like to investigate RQ2 and RQ3 as outlined earlier, to provide a comprehensive understanding and potentially required addition of fairness in today’s sharing economy platforms.

2 NOTION OF FAIRNESS IN SHARING ECONOMY PLATFORMS

In this section, we first present the system model of a matching platform. Then, we introduce the notions of fairness for such platforms.

2.1 System Model

We present the system model for the matching platform in Figure 2 to conceptualize the notions of fairness. The platform produces the sequence of matches between customers and providers over time. The matching decision at any time instant t can be represented as a tuple $\{C_i, P_j, t\}$ involving the customer-provider pair $\{C_i, P_j\}$.

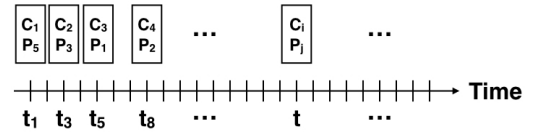


Figure 2: The matching platform produces the matching between different customers and providers over time.

Then, we define two functions over the tuple $\{C_i, P_j, t\}$ at time t :

- (i) **Utility function** $U(\{C_i, P_j, t\})$: A measure of utility derived by the customer C_i for a matching $\{C_i, P_j, t\}$, and
- (ii) **Benefit function** $B(\{C_i, P_j, t\})$: A measure of benefit received by the provider P_j for the matching $\{C_i, P_j, t\}$.

The exact form of utility function may vary from domain to domain. For example, for a renter in Airbnb, utility will depend on the size of the rooms, cleanliness, the location, or the price of the rental property; whereas, for an employer in Freelancer, the utility will be decided based on the qualification of the provider to carry out the task. Similarly, the exact benefit function for the providers may also vary depending on the domain, although the benefit will often map to the revenue of the provider earned by finishing the task. The customer utility may not be as explicit as the provider benefit.

For a customer request at any given time, the platform can produce a list of available providers ranked according to their relevance. The relevance corresponds to the predicted utility the matched provider would generate for the customer. Starting from the top ranked provider, the platform can then move down the list, deciding on the match by additionally considering other simultaneous and outstanding demands. While such an approach would maximize the utility for the customers (i.e., maximize $U(\{C_i, P_j, t\}), \forall t$), in platforms where a provider becomes available as soon as the formerly assigned task is complete, often only a few providers (e.g., the ones that happen to have the highest rating) would be matched again and again, resulting in the *starvation* of other providers in a rich-get-richer scheme. Such a scenario might lead to amplification of small differences in ratings between equally relevant providers, and would make it especially difficult for new, equally qualified providers, to join the system.

Since there is no fairness for the providers in the standard matching scheme we described, we propose two notions of provider fairness that the matching platform can attempt to guarantee.

2.2 Parity Fairness

A simple notion of fairness would be to maintain parity between the providers, i.e., **over time, the sum of received benefits of all providers should be similar**. This can be achieved by maintaining a virtual queue of all providers, and exposing them to the customers in a round-robin manner.

Advantages and limitations: Although the above notion guarantees that no provider is starved and they all accumulate similar benefits over the long term, it assumes all matches are equally relevant and is thus oblivious to customer utility. However, different customers have different service requirements, and hence they might gain no utility if a non-relevant provider is matched. For example, Uber offers the option to hire different cabs like hatchbacks, SUVs etc. Therefore, a passenger asking for a SUV would get zero utility if a hatchback driver is assigned to her. Similarly, in Freelancer some employers might need people with experience in Machine Learning, whereas others may want to hire people with background in Networking. In such cases, the employers would gain little utility if the platform does not match them to people with relevant expertise.

2.3 Proportional Fairness

Actual and deserved benefits

To control for the missed benefit opportunities of all relevant drivers, we introduce two types of benefits for a provider P_j :

- (i) **Actual benefit**, which is the sum of the benefits received by P_j from the matches made by the platform till time t , and
- (ii) **Deserved benefit**, which is the sum of benefits provider P_j would have made on all platform matches made until time t : $\{C_i, P_j, t_k\}$, $t_k < t$, scaled by the utility the customer C_i would have received on that match.

On each customer request, the platform predicts the customer utility (relevance) and provider benefit for every provider-customer pair. If a provider is matched to the customer, the platform updates the *actual benefit* of the provider. Otherwise, the missed benefit of the non-match scaled by the relevance (customer utility) is added to the *deserved benefit* of the provider. In other words, the deserved benefit accumulates all benefits which could have been accrued by the provider if she was matched to all requesting customers, scaled by the chance of matching expressed by the match relevance.

Proportionally fair sharing-economy platforms

Using the definitions of deserved and actual benefits provided above, we propose a new notion of fairness in sharing economy platforms: A platform is fair if **over time, the actual benefit is proportional to the deserved benefit for every provider**.

The proposed notions have a number of useful properties:

- (i) the deserved benefit of non-relevant providers does not increase on missed matches, (ii) the deserved benefit of equally relevant providers will increase at the same rate, (iii) the deserved benefit will increase proportionately to the potential benefit of each match, which enables one to control the actual and deserved benefit ratios.

In Uber, for example, we can think of the deserved benefit for drivers as the accumulation of potential benefits if they were matched with every passenger requesting a ride. However, the deserved benefit for a hatchback driver will not increase if the passenger asked for SUV. The deserved benefit will grow reverse proportionately

to the distance between the passenger's location and the driver's location, when location is used to determine relevance. When a driver is logged out of the system, her deserved benefit does not change. Moreover, as the deserved benefit for a provider will increase during her waiting time when logged-in to the system, the proposed definition ensures there is no starvation.

3 MAPPING TO PROCESSOR SCHEDULING

When looking at the problem of fairness in two-sided markets in a temporal way, we can draw an analogy to problems of fair sharing of constrained resource, such as a processor or network bandwidth. In our scenario, the customers requesting a match correspond to incoming packets, and the providers correspond to processors with limited capacity. The goal of the resource sharing algorithms is to spread the workload among the processors in a fair way.

The problem of fairness has been extensively studied in realtime systems and networking literature, where the ideal notion of fair scheduling is captured by **generalized processor sharing** (GPS) algorithms [29]. However, GPS algorithms make certain assumptions which do not hold in practice. For example, they assume that the traffic in a network is *fluid* and can be split at any arbitrary point. In reality, traffic comes in discrete packets. Analogously, in the context of sharing economy platforms, the matching between a provider and a consumer constitutes a discrete event.

To escape the unrealistic assumptions of GPS, alternative techniques such as **Weighted Fair Queuing** [2] have been proposed. An implementation of fair matching in sharing economy platforms could directly utilize the weighted fair queuing techniques. An especially promising direction is to use a priority based scheduler [27], which would generate the matching between providers and consumers depending on the priority value

$$PR_j = 1 - \frac{AB_j}{DB_j}$$

where AB_j and DB_j denote the actual and desired benefits of provider j . At time t , the matching platform should pick a provider with high relevance and high priority value.

Finally, utilizing previous work on **Hierarchical Fair Scheduling** [3], the proposed solution could be generalized to tackle the problem of group fairness. Recent works have shown that consumers tend to give biased evaluations and ratings based on providers' gender, race, and other protected attributes [13, 15]. This means that the utility function can be biased against some socially salient groups (e.g., women, blacks) and recreate inequality in the matching. Hierarchical Fair Scheduling would allow the platforms to implement group fairness on top of the individual fairness, and control for factors such as equal exposure of men and women, minimum exposure for a minority group, or closing the wage gap between different demographic groups.

4 CONCLUDING DISCUSSION

In this position paper, we have provided a systematic way to think about fairness in sharing economy platforms. The proposed definition of fairness incorporating the temporal dimension enables implementations that could draw from past works on fairness in processor scheduling.

While our focus in this paper was to consider fairness for the providers, there are remaining questions regarding the incentives of different stakeholders to participate in a fair matching framework.

Incentives for the providers: The obvious incentive for providers is the guarantee of getting a fair share of exposure, which translates to equal income opportunities. Additionally, there is an intrinsic feedback loop in the two-sided market systems – with more providers joining a fair system, the pool of customers can increase, guaranteeing a more steady demand for the providers.

Incentives for the customer: In many two-sided markets, matches of customers with a single most relevant provider do not occur. Rather, it is often the case that many providers are equally relevant to a request, and that otherwise the relevance scale is discrete (e.g., relevant, somewhat relevant, not relevant). For example, in cab-riding services like Uber, there is not much difference between the skillset of the providers, therefore all equally relevant providers should have an equal chance of being matched to customers.

Customers should also care about having a wide pool of providers available for the times when the demand is higher, and having fair matching algorithms may help keep providers inside the system. Alternatively, the platform might explicitly ask the customers to participate in the fair matching scheme by offering additional monetary incentives. For example, Uber might offer a cheaper fare if a passenger would be willing to wait longer to get the cab. There are research efforts on designing incentive strategies [20], which may be applicable in this context.

Incentives for the platform: The matching platforms in the two-sided markets thrive on attracting both providers and customers. With a wide pool of users on both sides, they will be more resilient to the loss of user interest. Interestingly, recent incidents like the *Delete Uber* movement [4] show that the users are motivated to boycott platforms due to unfairness, among other reasons. Therefore, building user trust and loyalty by guaranteeing fairness for different stakeholders is a reasonable strategy towards a long-term success of the company.

Overall, our work provides a novel way of thinking about fairness in sharing economy platforms. We hope that it will spark the research on fair matching implementations, and investigations into the guarantees that could be provided for different ecosystem participants.

Acknowledgments: The authors thank the anonymous reviewers whose suggestions helped to improve the paper. A. Chakraborty is a recipient of Google India PhD Fellowship and Prime Minister’s Fellowship Scheme for Doctoral Research, a public-private partnership between Science & Engineering Research Board (SERB), Department of Science & Technology, Government of India and Confederation of Indian Industry (CII).

REFERENCES

- [1] Ian Ayres, Mahzarin Banaji, and Christine Jolls. 2015. Race effects on eBay. *The RAND Journal of Economics* 46, 4 (2015).
- [2] Jon CR Bennett and Hui Zhang. 1996. WF/sup 2/Q: worst-case fair weighted fair queueing. In *IEEE INFOCOM*.
- [3] Jon CR Bennett and Hui Zhang. 1997. Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking (TON)* 5, 5 (1997).
- [4] Emma Brockes. 2017. Is it OK to use Uber now that Travis Kalanick has resigned? theguardian.com/technology/2017/jun/28/uber-travis-kalanick-should-i-delete-app. (2017).
- [5] Abhijnan Chakraborty, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. 2015. Can trending news stories create coverage bias? on the impact of high content churn in online news media. In *Computation and Journalism Symposium*.
- [6] Abhijnan Chakraborty, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. 2016. Dissemination Biases of Social Media Channels: On the Topical Coverage of Socially Shared News. In *AAAI ICWSM*.
- [7] Abhijnan Chakraborty, Johnnatan Messias, Fabricio Benevenuto, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. 2017. Who Makes Trends? Understanding Demographic Biases in Crowdsourced Recommendations. In *AAAI ICWSM*.
- [8] Le Chen, Alan Mislove, and Christo Wilson. 2015. Peeking Beneath the Hood of Uber. In *ACM IMC*.
- [9] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated Experiments on Ad Privacy Settings: A Tale of Opacity, Choice, and Discrimination. In *PETS*.
- [10] Benjamin G. Edelman, Michael Luca, and Dan Svirsky. 2017. Racial Discrimination in the Sharing Economy: Evidence from a Field Experiment. *American Economic Journal: Applied Economics* (2017).
- [11] Carla Green and Sam Levin. 2017. Homeless, assaulted, broke: drivers left behind as Uber promises change at the top. theguardian.com/us-news/2017/jun/17/uber-drivers-homeless-assault-travis-kalanick. (2017).
- [12] Juho Hamari, Mimmi Sjöklint, and Antti Ukkonen. 2016. The sharing economy: Why people participate in collaborative consumption. *Journal of the Association for Information Science and Technology* 67, 9 (2016).
- [13] Anikó Hannák, Claudia Wagner, David Garcia, Alan Mislove, Markus Strohmaier, and Christo Wilson. 2017. Bias in Online Freelance Marketplaces: Evidence from TaskRabbit and Fiverr. In *ACM CSCW*.
- [14] Tamar Kricheli-Katz and Tali Regev. 2016. How many cents on the dollar? Women and men in product markets. *Science Advances* 2, 2 (2016).
- [15] Michael Luca. 2016. Reviews, reputation, and revenue: The case of Yelp. com. (2016).
- [16] Hughston McBain. 1944. Are customers always right. *The Rotarian* (1944).
- [17] Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Walach, and Emine Yilmaz. 2017. Auditing Search Engines for Differential Satisfaction Across Demographics. In *ACM WWW Companion*.
- [18] Christian Sandvig, Kevin Hamilton, Karrie Karahalios, and Cedric Langbort. 2014. Auditing algorithms: Research methods for detecting discrimination on internet platforms. *Data and discrimination: converting critical concerns into productive inquiry* (2014).
- [19] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2017. Fairness in Package-to-Group Recommendations. In *ACM WWW*.
- [20] Adish Singla, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause. 2015. Incentivizing Users for Balancing Bike Sharing Systems.. In *AAAI*.
- [21] Ana Swanson. 2015. *Racial bias in everything: Airbnb edition*. The Washington Post. washingtonpost.com/news/wonk/wp/2015/12/12/racial-bias-in-everything-airbnb-edition.
- [22] Latanya Sweeney. 2013. Discrimination in online ad delivery. (2013).
- [23] Rudy Telles Jr. 2016. Digital Matching Firms: A New Definition in the “Sharing Economy” Space. *ESA Issue Brief* (2016).
- [24] Jacob Thebault-Spieker, Loren G. Terveen, and Brent Hecht. 2015. Avoiding the South Side and the Suburbs: The Geography of Mobile Crowdsourcing Markets. In *ACM CSCW*.
- [25] Siddharth Tiwari. 2017. ‘Underhand tactics’ boost sales in e-commerce. sundayguardianlive.com/investigation/9881-underhand-tactics-boost-sales-e-commerce. (2017).
- [26] Matt Vella. 2012. The mega trend that swallowed Silicon Valley. fortune.com/2012/10/03/the-mega-trend-that-swallowed-silicon-valley. (2012).
- [27] Ji Yang, Zhang Yifan, Wang Ying, and Zhang Ping. 2004. Average rate updating mechanism in proportional fair scheduler for HDR. In *IEEE GLOBECOM*.
- [28] Georgios Zervas, Davide Proserpio, and John W Byers. 2014. The rise of the sharing economy: Estimating the impact of Airbnb on the hotel industry. *Journal of Marketing Research* (2014).
- [29] Zhi-Li Zhang, Don Towsley, and Jim Kurose. 1995. Statistical analysis of the generalized processor sharing scheduling discipline. *IEEE Journal on Selected Areas in Communications* 13, 6 (1995).

Position Paper: Exploring Explanations for Matrix Factorization Recommender Systems

Bashir Rastegarpanah
Boston University
bashir@bu.edu

Mark Crovella
Boston University
crovella@bu.edu

Krishna P. Gummadi
Max Planck Institute for Software
Systems
gummadi@mpi-sws.org

ABSTRACT

In this paper we address the problem of finding explanations for collaborative filtering algorithms that use matrix factorization methods. We look for explanations that increase the transparency of the system. To do so, we propose two measures. First, we show a model that describes the contribution of each previous rating given by a user to the generated recommendation. Second, we measure the influence of changing each previous rating of a user on the outcome of the recommender system. We show that under the assumption that there are many more users in the system than there are items, we can efficiently generate each type of explanation by using linear approximations of the recommender system's behavior for each user, and computing partial derivatives of predicted ratings with respect to each user's provided ratings.

ACM Reference format:

Bashir Rastegarpanah, Mark Crovella, and Krishna P. Gummadi. 2017. Position Paper: Exploring Explanations for Matrix Factorization Recommender Systems. In *Proceedings of Workshop on Responsible Recommendation at RecSys 2017, Como, Italy, August 2017 (FATREC 2017)*, 4 pages. <https://doi.org/10.18122/B2R717>

1 INTRODUCTION

Recommender systems are taking on an increasing role in shaping the impact of computing on society, and it is consequently important to develop methods for explaining the recommendations made by such systems.

Among the many possible goals for explanation [6], we focus on user-oriented explanation (explanations that assume the system is fixed) rather than developer-oriented explanation (explanations that guide system development). Within the user-oriented domain, we focus on explanations that have as their goal *transparency*: providing the user with an understanding of how the system formulated a recommendation.

Among the broad class of recommender systems approaches, one can distinguish *neighborhood methods*, based on computing similarities between items or users, from *matrix factorization*, which assigns items and users to a latent space in which inner product captures the affinity of a user for an item. Neighborhood methods naturally lend themselves to explanation: witness Netflix's recommendations in which, for a given movie previously viewed, a set of recommended movies is proposed. In this context, the previously viewed movie is treated as an explanation.

Matrix factorization (MF) methods, on the other hand, can be more accurate than neighborhood methods [2], but pose a greater challenge from an explanation standpoint. The associated challenges include:

- Matrix factorization methods make use of the *entire* set of previous recommendations – over all users and items – in formulating a single recommendation for a given user.
- Matrix factorization methods solve a non-convex optimization via heuristic methods, whose functioning can be quite opaque.

Our current work investigates two sets of corresponding questions:

- (1) In a context where multiple items (and users) can be said to have contributed to forming a recommendation, what is the most meaningful or useful feedback to give a user to explain a single recommendation?
- (2) Given the complexity of MF approaches, are there approximate representations of the behavior of MF algorithms that we can use to construct such useful feedback?

A general strategy, exemplified by [4], provides some guidance in addressing the two questions. First, explanations should be in terms that are familiar to users: broadly, they should be in terms of features rather than in terms of, e.g., latent vectors. Second, useful explanations can be in terms of *interpretable* models – for example, decision trees or linear models – which can be chosen as *local approximations* of a more complex nonlinear model (such as a neural network).

In the remainder of this position paper we use this general strategy to address the questions above. Our general approach is via the use of *gradients* of the rating function, as in recent work on classifiers (e.g., [1, 5]). First, we propose gradient based metrics appropriate for MF recommender systems; then we describe how, in a certain commonly encountered scenario, one may approximately compute those gradients.

2 EXPLANATIONS

Assume x_{ij} indicates the rating given by user j to item i . To formulate an explanation for a given recommendation, we consider the case in which the system has given user j a recommendation for item i with an estimated rating of \hat{x}_{ij} . That is, the system has formed a prediction that user j will rate item i at \hat{x}_{ij} and has consequently proposed item i to the user.

In such a setting, user j may ask:

- (1) Which previous ratings have contributed the most to the predicted rating \hat{x}_{ij} ?

- (2) Which previous ratings have the most influence over the predicted rating \hat{x}_{ij} ? In other words, if things were different – e.g., different ratings had been provided in the past – which differences would matter most?

To answer these questions in terms familiar to users, it makes sense to use items and their ratings as the basic vocabulary (rather than, say, latent vectors). Furthermore, although an MF based recommender system implicitly takes into account the set of known ratings across all users and items in making its recommendation, other users' ratings are not under user j 's control. Hence it does not seem helpful to express our explanations in terms of ratings other than user j 's.

This leads us to propose the following kinds of explanations for MF recommender systems:

Impact We model each recommendation \hat{x}_{ij} in terms of known ratings given by the same user. That is, we formulate a model

$$\hat{x}_{ij} \approx \sum_{k \in R(j)} \alpha_k x_{kj}.$$

where $R(j)$ is the set of items that have been previously rated by user j , and we term $\gamma_k = \alpha_k x_{kj}$ the *impact* of known rating x_{kj} on the predicted rating \hat{x}_{ij} . The model is linear to support interpretability.

Influence In order to explain the influence of the known rating x_{kj} on the predicted rating \hat{x}_{ij} , we define:

$$\beta_k = \frac{\partial \hat{x}_{ij}}{\partial x_{kj}}$$

and we call β_k the *influence* of x_{kj} on \hat{x}_{ij} .

We envision the use of these quantities as an interface element of the recommender system. For any given recommendation \hat{x}_{ij} , the interface can present the highest impact ratings (those with largest γ_k) and the highest influence ratings (those with largest β_k), along with their values, as explanations for that recommendation.

3 ALGORITHMS

We now seek to find ways to compute approximations to impact and influence as defined in Section 2.

To start, we formalize the setting. Assume $\mathbf{X} \in \mathbb{R}^{m \times n}$ is a partially observed, real-value matrix containing user ratings. Each column is associated with a user and each row is associated with an item. An MF recommender system attempts to estimate unknown elements of the rating matrix. To do so, it finds factors $\mathbf{U} \in \mathbb{R}^{\ell \times m}$ and $\mathbf{V} \in \mathbb{R}^{\ell \times n}$ such that $\mathbf{U}^T \mathbf{V}$ agrees with the known positions in \mathbf{X} . The unknown ratings are then estimated by setting $\hat{\mathbf{X}} = \mathbf{U}^T \mathbf{V}$.

More specifically, the recommender system finds factors \mathbf{U} and \mathbf{V} by applying an algorithm \mathcal{A} to solve the following optimization problem:

$$\mathbf{U}, \mathbf{V} = \arg \min_{\tilde{\mathbf{U}}, \tilde{\mathbf{V}}} \sum_{(i,j) \in \Omega} (x_{ij} - \tilde{\mathbf{u}}_i^T \tilde{\mathbf{v}}_j)^2 \quad (1)$$

where Ω indicates the set of known entries in \mathbf{X} , $\tilde{\mathbf{u}}_i$ is column i of $\tilde{\mathbf{U}}$, and $\tilde{\mathbf{v}}_j$ is column j of $\tilde{\mathbf{V}}$.

3.1 U and V at a local optimum

To capture the effect of \mathcal{A} , let us define a function f such that for each user j , f returns the estimation of all item ratings for user j given the set of known item ratings of user j . In other words, f takes the column \mathbf{x}_j of the *observed* matrix \mathbf{X} as input, and returns the corresponding column $\hat{\mathbf{x}}_j$ of the *predicted* matrix $\hat{\mathbf{X}}$, i.e., $f(\mathbf{x}_j) = \hat{\mathbf{x}}_j$.

Now consider the properties of \mathbf{U} and \mathbf{V} at a local minimum of the objective function (1). In that case, each can be expressed as a linear function of \mathbf{X} . To see this, first note that for each user j we have $f(\mathbf{x}_j) = \mathbf{U}^T \mathbf{v}_j$. To capture the fact that only known entries matter in the solution of (1), we define \mathbf{W}_j to be a binary matrix with 1s on the diagonal in positions corresponding to the known entries of \mathbf{x}_j . Then it follows that \mathbf{v}_j is the least squares solution of

$$\mathbf{W}_j \mathbf{x}_j = \mathbf{W}_j \mathbf{U}^T \mathbf{v}_j$$

This implies that at a local minimum of (1), the following relationship holds between \mathbf{x}_j and $f(\mathbf{x}_j)$:

$$f(\mathbf{x}_j) = \mathbf{U}^T \mathbf{v}_j = \mathbf{U}^T (\mathbf{U} \mathbf{W}_j \mathbf{U}^T)^{-1} \mathbf{U} \mathbf{W}_j \mathbf{x}_j \quad (2)$$

3.2 A common case

To develop methods for approximating γ_k and β_k , we consider the case in which there are many more users in the system than there are items. For example, a movie recommendation system may have millions of users but only thousands of movies. In that case we formulate the following hypothesis:

HYPOTHESIS 1. Given matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, with $n \gg m$, let Ω be the set of known elements in \mathbf{X} and \mathcal{A} be an algorithm that computes \mathbf{U} and \mathbf{V} , a local optimum of (1). Assume we change element x_{ij} to x'_{ij} and rerun algorithm \mathcal{A} to find \mathbf{U}' and \mathbf{V}' , then \mathbf{U}' is approximately equal to \mathbf{U} and the only significant differences between \mathbf{V}' and \mathbf{V} lie in column j .

Informal justification for Hypothesis 1 is provided in Appendix A. We find that Hypothesis 1 holds consistently in empirical studies.

3.2.1 Influence. In cases where Hypothesis 1 holds, we can proceed as follows. We start by estimating influence. Our goal is to compute the Jacobian of the function $f(\cdot)$ evaluated at \mathbf{x}_j . That is, we seek:

$$\mathbf{J}^{(j)} = \frac{\partial f(\mathbf{x}_j)}{\partial \mathbf{x}_j}$$

We call $\mathbf{J}^{(j)}$ the influence matrix of user j .

Assume $\boldsymbol{\varepsilon}_i$ is a vector of size m in which element i is ε and all the other elements are zero. In order to compute each element of the influence matrix of user j , we need to compute function f at \mathbf{x}_j and at neighborhoods of \mathbf{x}_j that are defined by $\mathbf{x}_j + \boldsymbol{\varepsilon}_i$ for $i \in \{1, \dots, m\}$. Equation (2) provides a closed form formula of function $f(\cdot)$ when the input is one of the user rating vectors. Moreover, under Hypothesis 1 we know that equation (2) provides an approximation for $f(\cdot)$ when the input is a *modified* user rating vector in which only one of the elements is changed. Therefore we can state that Equation (2) holds not just at \mathbf{x}_j , but also within a small neighborhood around \mathbf{x}_j . Then:

$$\mathbf{J}^{(j)} = \frac{\partial f(\mathbf{x}_j)}{\partial \mathbf{x}_j} = \mathbf{U}^T (\mathbf{U} \mathbf{W}_j \mathbf{U}^T)^{-1} \mathbf{U} \mathbf{W}_j$$

Interestingly, the influence matrix of each user j only depends on U and on the set of items that have been previously rated by user j . In particular, it does not depend on the actual ratings that user j has given to any previous items. So if two users a and b happen to have rated exactly the same set of items, although the actual rating values may differ, their influence matrices $J^{(a)}$ and $J^{(b)}$ will be identical.

In summary, when Hypothesis 1 holds, then for a given user j and recommended item i , the influence of item k is $\beta_k = J_{ik}^{(j)}$.

3.2.2 Impact. Next, we turn to approximating impact. In section 2 we showed that the following linear model describes the output of an MF recommender system for user j as a function of his known ratings:

$$f(\mathbf{x}_j) = \mathbf{J}^{(j)} \mathbf{x}_j$$

where $\mathbf{J}^{(j)}$ is the influence matrix of user j . Therefore, the predicted rating for item i can be written as

$$\hat{x}_{ij} = \sum_{(k,j) \in \Omega} J_{ik}^{(j)} x_{kj}.$$

We define γ_k , the impact of known rating x_{kj} on predicted rating \hat{x}_{ij} as

$$\gamma_k = J_{ik}^{(j)} x_{kj}$$

which is simply the product of the influence of item k on the prediction for item i and the actual rating given by user j to item k .

We emphasize that our proposed method for computing γ_k is only one way of quantifying impact. In other words, one may choose another linear combination of known ratings of user j that results in \hat{x}_{ij} to define impact. While our method here has the interesting property that coefficients are identical to partial derivatives, one may choose another method to satisfy a different set of properties. For example, a recent work [5] studies the problem of attributing the prediction of a deep network to its input features. A similar approach can be adopted to define more elaborate measures of impact in the context of MF recommender systems.

4 EXAMPLE

To illustrate our proposal more concretely, we present an example using the MovieLens small dataset [3]. We choose the 650 most active users and the 50 most frequently rated movies. The resulting rating matrix has about 25% known entries. To this matrix we apply a well known matrix factorization algorithm (LMaFit [7]) with estimated rank 4 and obtain factors U and V .

If Hypothesis 1 holds, then as discussed above, if users a and b have rated the same set of items, changing the rating given by user a to item i ($x'_{ia} \leftarrow x_{ia} + \varepsilon$) and changing the rating given by user b to item i by the same amount ($x'_{ib} \leftarrow x_{ib} + \varepsilon$) should have an identical effect on the predicted ratings for all other items. In other words, we have:

$$f(\mathbf{x}_a + \varepsilon_i) - f(\mathbf{x}_a) = f(\mathbf{x}_b + \varepsilon_i) - f(\mathbf{x}_b) \quad (3)$$

To illustrate this, we find two users a (user 16) and b (user 211) who happen to have rated the same set of five movies in our data. Figure 1 (left side) shows the ratings given by these two users to these five movies. We then add 1 to user a 's rating for movie 4,

Table 1: Explanation for Terminator2

Rated movie	Influence
<i>Mission: Impossible</i> (1996)	5.00
<i>Twelve Monkeys</i> (a.k.a. 12 Monkeys) (1995)	1.01
<i>Star Wars: Episode IV - A New Hope</i> (1977)	-0.24
<i>Fargo</i> (1996)	-1.65
<i>Independence Day</i> (1996)	-2.74

rerun LMaFit, and compute the difference in predicted ratings for all movies. Next we repeat the same procedure, this time modifying only user b 's rating for movie 4. The two vectors of rating differences are shown on the right side of Figure 1, and we see that the changes across all movie ratings are nearly identical.

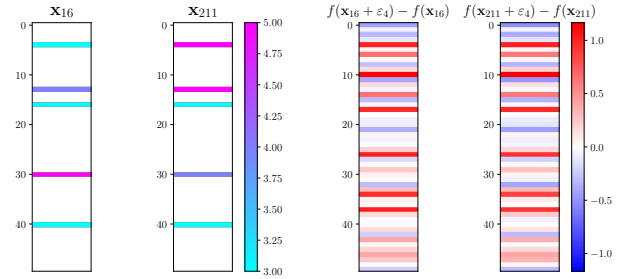


Figure 1: Users 16 and 211 have rated the same set of items. Left: original ratings; Right: changes to all predictions after modifying each user's rating for movie 4.

To illustrate the use of influence values in practice, we show in Table 1 a simple example drawn from our dataset. The table shows for user 16, the influence of each of the 5 movies that the user has rated on the system's predicted ratings for *Terminator2*. The Table shows that changing the user's previous ratings for *Star Wars* or *Fargo* would have much less influence on the predicted rating for *Terminator2* than would changing previous ratings for *Mission Impossible* or *Independence Day*.

5 CONCLUSION

In this position paper we've proposed two kinds of explanations for increasing the transparency of matrix factorization recommender systems: influence, and impact. We argue that these allow for interpretable responses to questions that are important to users: "What are the most important factors yielding this recommendation?" and "What are the factors whose change would most affect this recommendation?" The first question provides the users an understanding of how a recommendation is generated by the system based on the actions they have made in the past, while answering the second question provides the users with information that can be used to control the system's behavior in the future.

We have also shown that in the common case in which there are many more users than items (such as movie recommender systems), there are tractable computational approximations that can be used

to provide numerical values for influence and impact. Interestingly, we find that in this case influence is only determined by the set of movies rated, but not by the values of the ratings given.

We expect to develop these results in both theoretical and practical directions to explore the ultimate utility of these modes of explanation for matrix factorization recommender systems.

A JUSTIFICATION OF HYPOTHESIS 1

Here we present a justification for Hypothesis 1.

We consider the case in which $n \gg m$. In our analysis we make the assumption that a change to x_{ij} only results in changes to \mathbf{u}_i and \mathbf{v}_j (i.e., we focus on the first-order approximation to the effect of Algorithm \mathcal{A}). Let the updated latent vectors be \mathbf{u}'_i and \mathbf{v}'_j .

Intuitively, our argument is as follows. Updating \mathbf{u}_i to \mathbf{u}'_i results in changes to errors only in row i , and updating \mathbf{v}_j to \mathbf{v}'_j results in changes to errors only in column j . The effect of updates yielding \mathbf{u}'_i and \mathbf{v}'_j will generally attempt to decrease error at position (i, j) and will consequently tend to increase errors on other elements of row i and column j . Since there are many more elements in row i than there are in column j , an update to \mathbf{u}_i (to achieve a unit decrease in error at position i, j) will introduce more overall error than will an update to \mathbf{v}_j . Hence the bulk of change will occur in \mathbf{v}_j , while \mathbf{u}_i will remain relatively constant.

More formally, let $e_{ij} = (x_{ij} - \mathbf{u}_i^T \mathbf{v}_j)$ and $L = \sum_{ij} e_{ij}^2$. Before the change to element x_{ij} , the effect of \mathcal{A} has been to achieve $\frac{\partial L}{\partial \mathbf{u}_i} = \frac{\partial L}{\partial \mathbf{v}_j} = 0$. These partial derivatives are:

$$\frac{\partial L}{\partial \mathbf{u}_i} = -2 \sum_k e_{ik} \mathbf{v}_k \quad \frac{\partial L}{\partial \mathbf{v}_j} = -2 \sum_k e_{kj} \mathbf{u}_k \quad (4)$$

Now, we introduce a change to the rating in position (i, j) . Assuming that only \mathbf{u}_i and \mathbf{v}_j change during the subsequent optimization, then applying \mathcal{A} leads to:

$$\mathbf{u}'_k = \begin{cases} \mathbf{u}_i + \tilde{\mathbf{u}} & k = i \\ \mathbf{u}_k & k \neq i \end{cases} \quad \mathbf{v}'_k = \begin{cases} \mathbf{v}_j + \tilde{\mathbf{v}} & k = j \\ \mathbf{v}_k & k \neq j \end{cases} \quad (5)$$

Thus our goal becomes to establish that $\|\tilde{\mathbf{v}}\| \gg \|\tilde{\mathbf{u}}\|$.

Let e' be the new error values and L' be the new total squared error. At the new local optimum, we have $\frac{\partial L'}{\partial \mathbf{u}'_i} = \frac{\partial L'}{\partial \mathbf{v}'_j} = 0$.

$$\begin{aligned} \frac{\partial L'}{\partial \mathbf{u}'_i} &= -2 \sum_k e'_{ik} \mathbf{v}'_k & \frac{\partial L'}{\partial \mathbf{v}'_j} &= -2 \sum_k e'_{kj} \mathbf{u}'_k \\ &= -2(e'_{ij} \mathbf{v}'_j + \sum_{k \neq j} e'_{ik} \mathbf{v}_k) & &= -2(e'_{ij} \mathbf{u}_i + \sum_{k \neq i} e'_{kj} \mathbf{u}_k) \end{aligned} \quad (6)$$

Subtracting corresponding eqns in (6) and (4) and dropping factors of -2, we get:

$$\frac{\partial L'}{\partial \mathbf{u}'_i} - \frac{\partial L}{\partial \mathbf{u}_i} = e'_{ij} \mathbf{v}'_j - e_{ij} \mathbf{v}_j + \sum_{k \neq j} (e'_{ik} - e_{ik}) \mathbf{v}_k \quad (7)$$

$$\frac{\partial L'}{\partial \mathbf{v}'_j} - \frac{\partial L}{\partial \mathbf{v}_j} = e'_{ij} \mathbf{u}'_i - e_{ij} \mathbf{u}_i + \sum_{k \neq i} (e'_{kj} - e_{kj}) \mathbf{u}_k \quad (8)$$

Note that:

$$e'_{ik} - e_{ik} = -\tilde{\mathbf{u}}^T \mathbf{v}_k \quad k \neq j \quad (9)$$

$$e'_{kj} - e_{kj} = -\tilde{\mathbf{v}}^T \mathbf{u}_k \quad k \neq i \quad (10)$$

So substituting (9) and (10) into (7) and (8):

$$\frac{\partial L'}{\partial \mathbf{u}'_i} - \frac{\partial L}{\partial \mathbf{u}_i} = e'_{ij} \mathbf{v}'_j - e_{ij} \mathbf{v}_j + \sum_{k \neq j} (-\tilde{\mathbf{u}}^T \mathbf{v}_k) \mathbf{v}_k = 0 \quad (11)$$

$$\frac{\partial L'}{\partial \mathbf{v}'_j} - \frac{\partial L}{\partial \mathbf{v}_j} = e'_{ij} \mathbf{u}'_i - e_{ij} \mathbf{u}_i + \sum_{k \neq i} (-\tilde{\mathbf{v}}^T \mathbf{u}_k) \mathbf{u}_k = 0 \quad (12)$$

Now subtracting (12) from (11) we get:

$$e'_{ij}(\mathbf{v}'_j - \mathbf{u}'_i) - e_{ij}(\mathbf{v}_j - \mathbf{u}_i) + \sum_{k \neq i} (\tilde{\mathbf{v}}^T \mathbf{u}_k) \mathbf{u}_k - \sum_{k \neq j} (\tilde{\mathbf{u}}^T \mathbf{v}_k) \mathbf{v}_k = 0 \quad (13)$$

In (13), we note that the terms $e'_{ij}(\mathbf{v}'_j - \mathbf{u}'_i)$ and $e_{ij}(\mathbf{v}_j - \mathbf{u}_i)$ are small compared to the two summation terms. Therefore we can approximately argue:

$$\sum_{k \neq i} (\tilde{\mathbf{v}}^T \mathbf{u}_k) \mathbf{u}_k \approx \sum_{k \neq j} (\tilde{\mathbf{u}}^T \mathbf{v}_k) \mathbf{v}_k \quad (14)$$

$$\tilde{\mathbf{v}}^T \sum_{k \neq i} \mathbf{u}_k \mathbf{u}_k^T \approx \tilde{\mathbf{u}}^T \sum_{k \neq j} \mathbf{v}_k \mathbf{v}_k^T \quad (15)$$

This establishes a relationship between $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{u}}$. To make quantitative predictions, we can assume, e.g., that \mathbf{u}_k and \mathbf{v}_k are i.i.d. multivariate Gaussian random variables $\mathcal{N}(0, \Sigma)$ with $\Sigma = E[\mathbf{u}_k \mathbf{u}_k^T] = \sigma^2 I$. Then in expectation:

$$E \left[\tilde{\mathbf{v}}^T \sum_{k \neq i} \mathbf{u}_k \mathbf{u}_k^T \right] \approx E \left[\tilde{\mathbf{u}}^T \sum_{k \neq j} \mathbf{v}_k \mathbf{v}_k^T \right] \quad (16)$$

$$\tilde{\mathbf{v}}^T \sum_{k \neq i} E[\mathbf{u}_k \mathbf{u}_k^T] \approx \tilde{\mathbf{u}}^T \sum_{k \neq j} E[\mathbf{v}_k \mathbf{v}_k^T] \quad (17)$$

$$(m-1)\sigma^2 \tilde{\mathbf{v}}^T \approx (n-1)\sigma^2 \tilde{\mathbf{u}}^T \quad (18)$$

So we have that $\|\tilde{\mathbf{v}}\| / \|\tilde{\mathbf{u}}\| \approx (n-1)/(m-1)$.

REFERENCES

- [1] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to Explain Individual Classification Decisions. *J. Mach. Learn. Res.* 11 (Aug. 2010), 1803–1831. <http://dl.acm.org/citation.cfm?id=1756006.1859912>
- [2] Yehuda Koren and Robert Bell. 2011. Advances in Collaborative Filtering. In *Recommender Systems Handbook*. Springer, 145–186.
- [3] MovieLens [n. d.]. MovieLens dataset. <https://grouplens.org/datasets/movielens/>. ([n. d.]).
- [4] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why Should I Trust You?: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1135–1144.
- [5] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. *CoRR* abs/1703.01365 (2017). <http://arxiv.org/abs/1703.01365>
- [6] Nava Tintarev and Judith Masthoff. 2011. Designing and evaluating explanations for recommender systems. In *Recommender Systems Handbook*. Springer, 479–510.
- [7] Zaiwen Wen, Wotao Yin, and Yin Zhang. 2012. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation* 4, 4 (2012), 333–361. <https://doi.org/10.1007/s12532-012-0044-1>

Do News Consumers Want Explanations for Personalized News Rankings?

Maartje ter Hoeve*
University of Amsterdam
Amsterdam, The Netherlands
maartje.terhoeve@student.uva.nl

Anne Schuth
Blendle
Utrecht, The Netherlands
anneschuth@blendle.com

Mathieu Heruer
Blendle
Utrecht, The Netherlands
mathieu@blendle.com

Martijn Spitters
Blendle
Utrecht, The Netherlands
martijn@blendle.com

Daan Odijk
Blendle
Utrecht, The Netherlands
daan@blendle.com

Maarten de Rijke
University of Amsterdam
Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

To gain more insight in the question whether personalized news recommender systems should be responsible for their recommendations and transparent about their decisions, we study whether news consumers want explanations of why these news articles are recommended to them and what they find the best way to explain this. We survey users of Blendle's news recommendation system, and from 120 respondents we learn that news consumers do want explanations, yet do not have a very strong preference for how explanations should be shown to them. Moreover, we perform an A/B test that shows that the open rate per user does not change if users are provided with reasons for the articles recommended for them. Most likely this is because users did not pay attention to the reasons.

CCS CONCEPTS

•**Information systems** → **Personalization**; *Recommender systems*; *Relevance assessment*; **Presentation of retrieval results**;

KEYWORDS

News recommendation; Transparency; Explainable models

1 INTRODUCTION

The European Union has approved the General Data Protection Regulation (GDPR) on April 14, 2016. The GDPR will be enforced on May 25, 2018, and states, amongst others, that one needs to be able to explain algorithmic decisions. At the time of writing (mid 2017), the broader implications of this regulation are not clear, but there does seem to be a broadly accepted view that citizens in a transparent society are entitled to explanations of technology-driven processes, especially as algorithmic decisions increasingly influence our daily life. To which degree do citizens actually care about this? That is, are people who base their decisions and lives

on the outcomes of algorithmic decisions, interested in receiving information on why a decision was made for them?

One area in which transparency and explainability are particularly important is *news*, both concerning news content and concerning the technology used to expose citizens to news (e.g. [2, 5, 11]). We focus on one aspect of technology that helps to expose citizens to news: news search and recommendation. Increasingly, news consumers use personalized services to consume news, often based on algorithmic or mixed algorithmic/editorial selections (e.g. [4, 6]). These personalized services determine to a large extent what news items their consumers read. It is tempting to state that these services should take their responsibility and be transparent about their choices by explaining their decisions to their users. However, do consumers of personalized news services care about explanations of the way in which their personalized selections were determined? We study this question in the setting of Blendle,¹ a Dutch start-up backed by amongst others The New York Times. Every day, Blendle users receive a personalized selection of news articles, selected based on a number of features that capture their reading behavior and topical interests. On top of this, Blendle users also receive a number of *must reads* every day; these articles are selected by Blendle's editorial staff and are the same for everyone. This is one of the ways to prevent users ending up in their own filter bubble. Blendle allows users to purchase a single news article instead of having to buy an entire newspaper (using micropayments) or to pre-pay via a subscription for their personal selection (called Blendle Premium). Users have the possibility to receive a refund for an article if they are not satisfied with it.

We have three research questions. Firstly, we investigate whether users would like to see explanations about why they see the articles selected for them. Secondly, we study what users find the best way to receive these explanations. Thirdly, we would like to know whether users open more articles if they are provided with explanations. In answering these research questions, our findings contribute to our understanding of the urge that news consumers feel to read articles from a transparent news recommender system, and because of this, to what extent news recommender systems should be accountable for their decisions. More broadly, our findings contribute to our understanding of how explainability can be operationalized.

*Research performed while intern at Blendle.

Additional authors: Ron Mulder (Blendle, ron@blendle.com), Nick van der Wildt (Blendle, nick@blendle.com).

This article may be copied, reproduced, and shared under the terms of the Creative Commons Attribution-ShareAlike license (CC BY-SA 4.0).

FATREC 2017, Como, Italy

© 2017 Copyright held by the owner/author(s). ...\$15.00

DOI: 10.18122/B24D7N

¹<http://www.blendle.com>

2 RELATED WORK

Tintarev and Masthoff [12] list seven possible aims when explaining the outcomes of an algorithm to users: *transparency*, *scrutability*, *trust*, *effectiveness*, *persuasiveness*, *efficiency* and *satisfaction*. Vig et al. [13] describe two explanation styles: *justifications* and *descriptions*. Justifications are focused on providing conceptual explanations that do not necessarily expose the underlying structure of the algorithm, whereas descriptions are meant to do exactly that. Several studies have investigated the explainability of recommender systems and the effects of adding explanations to the system (e.g. [1, 3, 8–10]). A number of these studies use *collaborative filtering* as recommendation technique [1, 3]. Collaborative filtering has been proven to be difficult to use for news recommendations due to what is known as the *cold start* or *first rater problem* [7, 14]. I.e., a news article needs to be recommended right after its release. At that moment the article has not been read yet and for this reason no information that can be used for collaborative filtering is known yet. In particular, Herlocker et al. [3] investigate the addition of explanations to the recommender system of *MovieLens*, that uses collaborative filtering as its recommendation technique. Users of *MovieLens* answer positively to the question whether they would like to see explanations added to the recommender system. This study differs from our study in its domain (i.e. news recommendations as opposed to movie recommendations), the underlying recommender system and because of that, the explanations that can be used (the aforementioned collaborative filtering) and it dates from the year 2000, whereas the recommender system research field has not been static since then. Several studies show that users are sensitive to the way explanations are shown [1, 9]. E.g., Bilgic and Mooney [1] find that users are more accurately able to decide which items are relevant for them based on “key-word style” explanations (a content based approach: which other items they interacted with before contain similar words) than on “neighbourhood style” explanations (how similar people rated this particular item).

3 RESEARCH QUESTIONS AND DESIGN

We address the following research questions: (RQ1) Do users want to receive explanations why particular news items are recommended to them? (RQ2) What way of showing news recommendations do users prefer? (RQ3) Do users open more articles if we provide explanations of why users see these articles? To answer these research questions, we design two experiments: a user study to answer RQ1 and RQ2 and an A/B test to answer RQ3. Both are detailed below.

3.1 User study

Our user study investigates whether users find it helpful to receive explanations about why particular news articles are selected for them and how they would like to see these explanations.

We designed five different types of reasons to explain our recommendations, to be judged by participants in the study. Table 1 summarizes all five reason types. Visible reasons are reasons that can be found on the card (e.g., the topic or the length of the article), invisible reasons are reasons that cannot be found on the card itself (e.g., the author). Figure 1 shows examples of items that were shown to participants.

Table 1: Reason types used in the user study.

Reason type	Example
1. Single reason, visible	Because you like politics
2. Single reason, invisible	Because you like this author
3. Multiple reasons, visible	Because you like politics and long articles
4. Multiple reasons, combined	Because you like <i>De Tijd</i> and this author
5. Bar chart	See Figure 1e

5+ De reden waarom ik dit artikel zie, geeft mij vertrouwen in het algoritme dat dit artikel voor mij selecteert.



Figure 2: Example interface of the questionnaire, for a single question. Judgment at the top (Q4, see Table 2).

We sent out an email questionnaire to a selection of Blendle users, 541 in total. Approximately two third of these users had a Blendle Premium subscription at the time of sending. The rest of these users used the micropayment system, but had used Blendle Premium at least once, for example via a free trial that lasted for one week.

Participants were shown three different types of explanations (“reason types”) and subsequently asked to answer five questions per type. To limit the length of the survey, participants are asked to judge three types of explanations, out of the five described above. Figure 2 shows an example of the interface of the questionnaire. To make sure the results are not biased by the type or content of an article, three different articles were considered: 179 users were sent the first article, 180 users were sent the second, and 182 users were sent the third article.

Note that users were not sent the entire article, but only the introduction card to the article. This article card contains a picture, a brief introduction to the article, the name of the newspaper or the magazine, a topic, the approximate reading time of the article, how many people liked the article and the reason type. The card functions to give the news consumer a brief introduction to the article

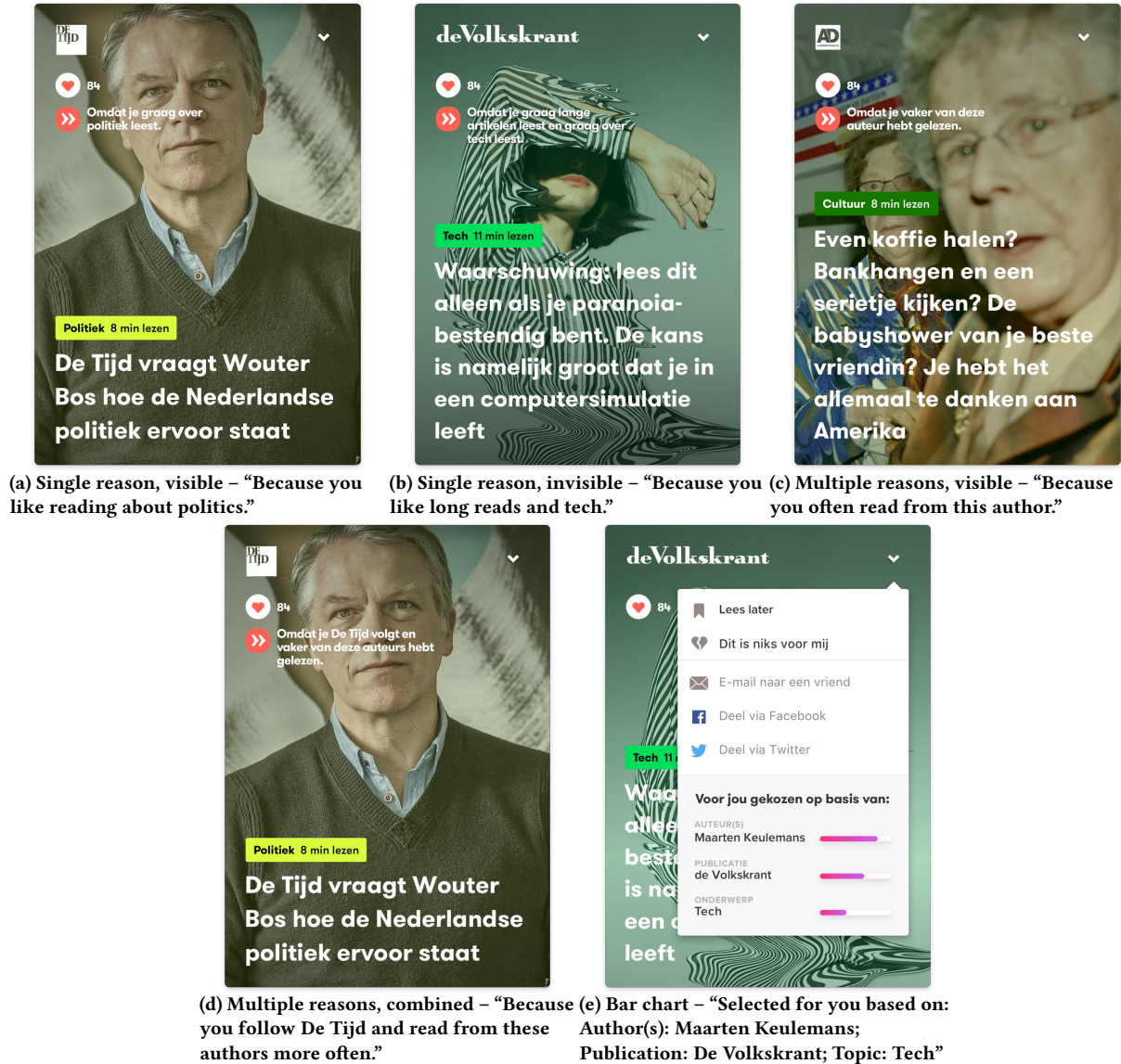


Figure 1: Examples of reason types as shown to users in our user study. Textual reasons are in the lines that start with “Omdat” (*because*). For the bar chart layout the reasons starts with “Voor jou gekozen” (*selected for you*). Translations are given below each article.

Table 2: Questions used in the questionnaire as part of our user study.

Type	Question asked (English translations of the Dutch questions)
Q1. Wants reasons?	On the figure below you can see what an article currently looks like on Blendle Premium. The articles that you see are chosen based on your personal preferences and what you like to read. Imagine we would give you more information about why we chose a certain article for you. Would you find that useful?
Q2. Transparency	I understand the way that is used to explain why I see this article.
Q3. Sufficiency	I get enough information to decide whether I would like to read this article.
Q4. Trust	The reason why I see this article, makes me trust the algorithm that selected this article for me.
Q5. Satisfaction	I am satisfied with the way in which this article is shown to me.

to decide whether he or she would like to read it. Figures 1a, 1b, 1c show the three different types of article cards that are used. Note that users are randomly divided over all three article types and over reason types. That is, no personalization was used here. We did not, however, completely randomize the order in which participants answer questions. First, users are either shown reason type 1 or 2, then 3 or 4. All users are shown reason type 5, as reason type 5 is very different from the other reason types. In three final questions participants are asked to fill in their age and gender and whether they would like to add some final remarks (if any).

The questions that were asked for each participant are detailed in Table 2. First, we ask participants whether they would find explanations useful and we ask them to choose between *yes*, *somewhat*, *no* or *I don't know* as possible answers. We then show several examples of explanations and ask participants to judge the examples on four Tintarev and Masthoff [12]'s dimensions: *transparency*, *sufficiency*, *trust* and *satisfaction*, all on a five point scale. We decided to omit questions on Tintarev and Masthoff [12]'s *scrutability*, *efficiency* and *effectiveness* as metrics at this stage of our research, as participants are not confronted with their own personal selection of news. For this reason, they will not be able to reliably judge whether they would open this article. Note that if participants have selected *no* or *I don't know* as a reply to whether they would like explanations, we tell them we would still like to show them some possible ways of explaining their articles and ask for their judgment.

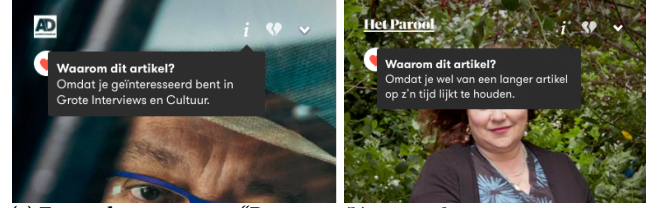
3.2 A/B test

In order to investigate whether users open more articles when they are provided with reasons of why they see these articles, we perform an A/B-test with two groups of Blendle users. Users are randomly assigned to a group. One of these groups is provided with explanations for the articles they see. The other group does not receive any explanations. Both groups are real Blendle users, i.e., we do not use an artificial experimental setting, but use the every day Blendle environment. The reasons shown to users in the "treatment group" are created heuristically. That is, we use a *justification* instead of an actual *description* in the sense of [13]. In our experiment, we use textual justifications. Two examples are given in Figure 3; the justifications are provided at the top of the article card, in the black boxes that pop up once a user has hovered over the "i" icon. This is different from the reasons tested in the user study, as we decided to launch a change in design that was as small as possible. All reasons are given in Table 4.

We run the A/B test for 24 days on 100% of our users.² As our objective, we measure the open rate, per day in each group.

In this study we define *open rate* as the *number of reads* over the *number of users*. We define the *number of reads* as the number articles that are opened by a user, without them asking for a refund. If users open an article multiple times (over any number of days), we only count the first time. The *number of users* is defined as the number of unique users that viewed their selection.

We test for differences in open rate between the two groups using a two-tailed paired t-test with $\alpha = 0.05$. Samples from both groups on one day form a pair. We discretize by days as news consumption



(a) Example reasons 1 – “Because you are interested in long interviews and Culture”.

(b) Example reasons 2 – “Because you seem to like longer articles”.

Figure 3: Example reason types used during the A/B test.

Table 3: Participant answers to Q1: Would you like to see more information on why articles are selected for you?

User wants reasons	Times answered
Yes	65
Somewhat	24
No	26
I don't know	5

varies over time. For the “reason group” we also count whether users have actively seen reasons, that is, hovered over the “i” icon. Moreover, we track whether users have seen reasons within two minutes before opening the article and if so, which reason that was.

4 RESULTS AND DISCUSSION

Here we answer our research questions. The first two questions are answered in Sections 4.1 and 4.2 by analyzing the results of our user study. In Section 4.3 we use the results of our A/B test to answer the last question.

A total of 120 users filled out our survey, of which 41 answered questions about the first article type, 36 about the second and 43 about the third article type. Of these 120 users, 103 users had a Blendle Premium subscription, while 17 users used the micropayment system at the time of sending out the survey. As there are not enough responses of non-premium users to put them in a separate group, we perform our analysis on all respondents together.

4.1 Do users want recommendation reasons?

Table 3 shows the results of what users answered to the question whether they would like to see better explained why they see articles in their selection. The significant majority answered *yes* or *somewhat* to this question, if compared to the number of people that answered *no* or *I don't know* ($\chi^2 = 14.55, p < 0.001$).

4.2 Do users want a particular type of recommendation reasons?

Table 6 shows the total average and standard deviation on all three articles combined, as well as the mean and standard deviation per question per article. Table 7 shows whether the differences in scores for the different types of questions are statistically significant or not. As the answers are independent, yet not necessarily sampled from the normal distribution, we use the two-sided Mann-Whitney U test, with $\alpha = 0.05$ as significance level. The sample sizes can be found in Table 5. From these results a few points stand out. First of all, although users do want more information about why they

²For competitiveness reasons we cannot reveal the size of the control and treatment groups.

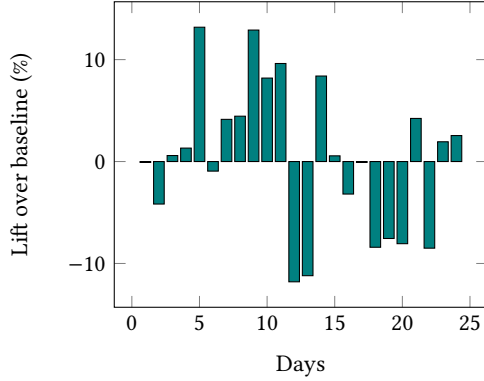


Figure 4: Lift in open rate for the group with recommendation reasons over the baseline without reasons.

see a certain article, the results do not show a clear preference as to which type of explanation users prefer. Only a few differences were significant (shown in boldface in Table 7). However, when we correct for the number of comparisons that we make, and take $\alpha = 0.001$ as significance level (using the Bonferroni correction and dividing our original α by 50, the number of comparisons that we make), none of the reason types scores significantly higher than another reason type. Another interesting point to make is that the standard deviations of the scores on the fifth reasoning type are, on average, bigger than the standard deviations of the scores on the other reasoning types, i.e., users either seem to like this way of showing reasons, or they do not.

4.3 Do users open more articles when provided with explanations?

In our A/B test, after 24 days, we see that users that were shown the recommendation reasons (the “reason group”) have a lift in open rate of 0.33%. This difference is plotted in Figure 4 and is not significant ($t = -0.29, p = 0.77$).

Of all individual users in the reason group, 9.8% has seen at least one recommendation reason. Of all users who opened an article, 1.08% had seen the recommendation reason within two minutes before they opened that particular article. These users saw 1.27 reasons on average, with a standard deviation of 0.73. That is, not many users saw the reasons, which explains why we do not observe a difference in open rate per user between the two groups. Different, more prominent designs, may yield different results.

Figure 5 shows how often users saw each particular reason, in comparison to the total number of times users saw a reason. Reason type 6 is seen most often. This is the explanation that is given for the *must-reads*, i.e., not based on any form of personalization. These must-reads are on top of the user’s page, which can bias these results.

5 CONCLUSION

In this study we investigated whether news consumers would like to receive explanations about why articles were selected for their personalized selections of news articles. We also investigated how they would prefer to receive these explanations. Moreover, we studied whether news consumers open more articles, if they are provided with reasons.

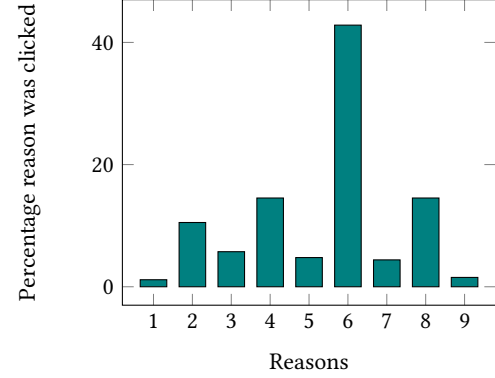


Figure 5: Reasons clicked before opening the article, see Table 4 for mapping.

Table 4: Reason mapping for reasons used in A/B test.

Number	Reason
Reason 1	Because you often read about TOPIC
Reason 2	Because you are interested in TOPIC
Reason 3	Because we think NEWSPAPER could be interesting for you
Reason 4	The editors really liked this piece
Reason 5	Because you follow NEWSPAPER
Reason 6	According to the editors, this is one of the best stories of the day. No matter your preferences
Reason 7	Because you often read from NEWSPAPER
Reason 8	Because you seem to like a long read every now and then
Reason 9	Because you often read from AUTHOR

Table 5: Sample sizes per reason type

Type 1	Type 2	Type 3	Type 4	Type 5
66	56	63	55	120

Our questionnaire showed that a large majority of the respondents would like to receive these explanations, yet they do not show a clear preference as to how they would like to see these. Our A/B test shows that the open rate per user does not increase by adding explanations. In fact, in many cases, users do not read the explanations.

More broadly, our research shows that users nowadays still attach importance to explanations of algorithmic decisions broader than the domain described in [3] and it motivates us to strive for transparent, responsible and accountable recommender systems.

Even though we tested several designs for explanations in our questionnaire, the number of options that we were able to expose to our participants was limited. It could very well be that alternative designs would be preferred by news consumers.

Hence, as future work, we recommend that A/B tests with additional designs are conducted. They may either result in a clearer preference for a particular way of explaining recommendations or further strengthen our conclusions. We especially recommend conducting A/B tests with reasons clearly visible, that is, not behind an icon as in the work reported here. More research in different domains, with different user groups, should lead to insights into the generalizability of our findings.

Table 6: Mean and standard deviations of the scores on different types of judgments in the user study. The “reason types” refer back to the types of reason listed in Table 1.

Reason type	1		2		3		4		5	
Question	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Transparency	3.697	1.141	3.786	1.129	3.587	1.107	3.873	1.096	3.650	1.339
Sufficiency	3.530	1.076	3.625	1.028	3.333	1.098	3.764	0.953	3.408	1.275
Trust	3.000	1.115	3.250	1.122	3.032	1.023	3.400	0.984	3.500	1.258
Satisfaction	3.606	0.919	3.661	0.969	3.317	1.096	3.582	1.073	3.233	1.327
Average	3.458	0.798	3.580	0.836	3.317	0.916	3.655	0.798	3.448	1.154

Table 7: Statistical differences between reason types, between different questions.

	Type 2	Type 3	Type 4	Type 5
Type 1				
Transparency	$U = 1811.0, p > 0.05$	$U = 2059.0, p > 0.05$	$U = 1597.0, p > 0.05$	$U = 3858.0, p > 0.05$
Sufficiency	$U = 1868.0, p > 0.05$	$U = 2147.5, p > 0.05$	$U = 1571.5, p > 0.05$	$U = 4005.0, p > 0.05$
Trust	$U = 1860.5, p > 0.05$	$U = 2016.5, p > 0.05$	$U = 1512.0, p > 0.05$	$U = 3001.0, p < 0.05$
Satisfaction	$U = 1748.0, p > 0.05$	$U = 2347.0, p > 0.05$	$U = 1740.0, p > 0.05$	$U = 4304.0, p > 0.05$
Average	$U = 1684.0, p > 0.05$	$U = 2257.5, p > 0.05$	$U = 1591.0, p > 0.05$	$U = 3848.0, p > 0.05$
Type 2				
Transparency		$U = 1838.5, p > 0.05$	$U = 1422.5, p > 0.05$	$U = 3404.0, p > 0.05$
Sufficiency		$U = 1899.5, p > 0.05$	$U = 1397.0, p > 0.05$	$U = 3529.0, p > 0.05$
Trust		$U = 1938.0, p > 0.05$	$U = 1472.0, p > 0.05$	$U = 2900.0, p > 0.05$
Satisfaction		$U = 2038.5, p > 0.05$	$U = 1523.0, p > 0.05$	$U = 3734.5, p > 0.05$
Average		$U = 2041.0, p > 0.05$	$U = 1493.0, p > 0.05$	$U = 3505.0, p > 0.05$
Type 3				
Transparency			$U = 1417.5, p > 0.05$	$U = 3476.0, p > 0.05$
Sufficiency			$U = 1324.0, p < 0.05$	$U = 3472.0, p > 0.05$
Trust			$U = 1411.5, p > 0.05$	$U = 2847.5, p < 0.05$
Satisfaction			$U = 1441.5, p > 0.05$	$U = 3657.5, p > 0.05$
Average			$U = 1369.5, p < 0.05$	$U = 3416.5, p > 0.05$
Type 4				
Transparency				$U = 3469.5, p > 0.05$
Sufficiency				$U = 3676.0, p > 0.05$
Trust				$U = 2992.0, p > 0.05$
Satisfaction				$U = 3586.0, p > 0.05$
Average				$U = 3575.0, p > 0.05$

Acknowledgments. This research was supported by Ahold Delhaize, Amsterdam Data Science, the Bloomberg Research Grant program, the Criteo Faculty Research Award program, the Dutch national program COM-MIT, Elsevier, the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Microsoft Research Ph.D. program, the Netherlands Institute for Sound and Vision, the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.001.116, HOR-11-10, CI-14-25, 652.002.001, 612.001.551, 652.001.003, and Yandex. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

- [1] M. Bilgic and R. J. Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI*, volume 5, page 153, 2005.
- [2] N. Diakopoulos. Algorithmic accountability: Journalistic investigation of computational power structures. *Digital Journalism*, 3(3):398–415, 2015.
- [3] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.
- [4] I. Ilievski and S. Roy. Personalized news recommendation based on implicit feedback. In *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*, pages 10–15. ACM, 2013.
- [5] M. Karlsson. The immediacy of online news, the visibility of journalistic processes and a restructuring of journalistic authority. *Journalism*, 12(3):279–295, 2011.
- [6] J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 31–40. ACM, 2010.
- [7] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Aaai/iaai*, pages 187–192, 2002.
- [8] C. Musto, F. Narducci, P. Lops, M. De Gemmis, and G. Semeraro. Explod: A framework for explaining recommendations based on the linked open data cloud. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 151–154. ACM, 2016.
- [9] P. Pu and L. Chen. Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems*, 20(6):542–556, 2007.
- [10] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.
- [11] J. B. Singer. Contested autonomy: Professional and popular claims on journalistic norms. *Journalism studies*, 8(1):79–95, 2007.
- [12] N. Tintarev and J. Masthoff. A survey of explanations in recommender systems. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pages 801–810. IEEE, 2007.
- [13] J. Vig, S. Sen, and J. Riedl. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 47–56. ACM, 2009.
- [14] E. Vozalis and K. G. Margaritis. Analysis of recommender systems algorithms. In *The 6th Hellenic European Conference on Computer Mathematics & its Applications*, pages 732–745, 2003.

Presenting Diversity Aware Recommendations:

Making Challenging News Acceptable

Nava Tintarev

Delft University of Technology

n.tintarev@tudelft.nl

ABSTRACT

Recommender systems find relevant content for us online, including the personalized news we increasingly receive on Twitter and Facebook. As a consequence of personalization, we increasingly see content that agrees with our views, we cease to be exposed to views contrary to our own. Both algorithms and the users themselves filter content, and this creates more polarized points of view, so called “filter bubbles” or “echo chambers”. This paper presents a vision of a *diversity aware recommendation model*, for the selection and presentation of a diverse selection of news to users. This diversity aware recommendation model considers that: a) users have different requirements on diversity (e.g., challenge-averse or diversity seeking), and that b) items will satisfy these requirements to different extents (e.g., liberal or conservative news). By considering both item and user diversity this model aims to maximize the amount of diverse content that users are exposed to, without damaging system reputation.

CCS CONCEPTS

• **Information systems** → **Decision support systems**; • **Human-centered computing** → **Human computer interaction (HCI)**;

KEYWORDS

diversity, explanations, user-centered design

ACM Reference format:

Nava Tintarev. 2017. Presenting Diversity Aware Recommendations. In *Proceedings of The FATREC Workshop on Responsible Recommendation, Como, Italy., August 31, 2017*, 4 pages. <https://doi.org/doi:10.18122/B2HQ41>

1 INTRODUCTION

Recommender systems find relevant content for us online, including the personalized news we increasingly receive on Twitter and Facebook [18, 28]. As a result of personalization, we increasingly see content that agrees with our views, we cease to be exposed to views contrary to our own. Both algorithms and the users themselves filter content, and this creates more polarized points of view, so called “echo chambers” [2, 6, 20]. Recommender systems have both the potential to increase the diversity of content and narrow it. Over time using recommender systems has been found to slightly decrease the diversity of content that users consume [19]. However, Flaxman et al. found evidence that recent technological changes both increase and decrease various aspects of the partisan divide.

This article may be copied, reproduced, and shared under the terms of the Creative Commons Attribution-ShareAlike license (CC BY-SA 4.0).

The FATREC Workshop on Responsible Recommendation, August 31, 2017, Como, Italy.

© 2017 Copyright held by the owner/author(s).

DOI: [doi:10.18122/B2HQ41](https://doi.org/doi:10.18122/B2HQ41)

This suggests that there may be design choices for recommender systems that could decrease polarization.

In response to the issue of “echo chambers” and “filter bubbles”, this paper therefore introduces a *diversity aware recommendation model* for selecting and presenting a diverse selection of news to people. The goal is to develop presentational strategies for recommendations that consider both item and user diversity, and maximize the diversity of recommendations given to a person, without losing trust, or polarizing their existing opinions. A complete solution to the problem needs to consider *both* the biases that algorithms and humans bring to the filtering process. If item selection and presentation is done in too simple a way, for example, by suggesting articles that the person strongly disagrees with, they simply might not return to the system, or become more extreme in their views [8, 11]. Building on advances in understanding both the influence of *user diversity* and *item diversity* on perceived recommendation quality and perceived diversity, it is now possible to address filter bubbles from a new angle. The proposed approach is to study people’s *perceptions* of diversity using live experiments. Experimentation with people makes it possible to identify how *presentational strategies* can be used to manage the perceptions resulting from user and item diversity. These experimental findings will inform the new *diversity aware recommendation model*, which will allow us to expose people to a wider range of content, while maximizing their acceptance as much as possible for (for them) challenging content. By doing so, this research has the potential to decrease polarization of views.

2 RELATED WORK

Previous work focusing on mitigating filter bubbles can be divided into two approaches: better understanding of candidate items, and diversity aware recommendation algorithms. The first approach is reaching a point of maturity where it can start to support the second: helping users understand the candidate items through presentational strategies may mitigate the effects of challenging content presented to users through diversity aware algorithms.

Understanding the candidate items. The first approach is to help users to better understand the recommended items relative to a wider set of candidate items. Taking this approach, we have found that helping users control which people contributed to their information feed on Twitter increased their sense of transparency and control [14, 23]. However, we also found that users had a poor mental model for the degree of novel content discovered when presented with non-personalized tweets, and thus potentially more challenging information. We also found that visualizing users’ blind-spots, i.e., underexplored areas in the search space, encouraged them to explore these parts of the item space (under review). In this regard, the work of Nagulendra and Vassileva is also pertinent, finding

that visualisation increased understandability of the filtering mechanism [17]. This approach of better understanding the candidate is also underpinned by studies on visualizations [27], explanations [25], and critiquing [15], in recommender systems.

Diversity aware algorithms. The second approach is to develop diversity aware recommendation algorithms, or algorithms that address the risks of filter bubbles and polarization. To support discovery, news recommender systems need to strike a delicate balance between diversity and relevance: to find news articles that are diverse, and still highly relevant to a user. To increase relevance the computation of similarity (e.g., between items, users, or their ratings) has been the basis of recommendation algorithms. The challenge is thus to define similarity in a way that maintains relevance while sufficiently diversifying items in recommendation sets (c.f., [1, 30, 31]). Overall, item diversity has been successfully implemented before: diversified recommendations have been found to increase user satisfaction [31], helped users find target items faster [4], and increased the novelty of the items that are recommended [30]. Furthermore, while *perceived* novel content discovery contributes to the attractiveness of recommendations, diversification that is not mediated by perceived discovery has been found to *reduce* the attractiveness of recommendations [7]. More recent approaches (c.f., [13, 29]) have considered how re-ranking can be used to include diversity in an optimization function. However, these measures of diversity are still not well understood from a user-centered perspective, especially when dealing with *human perceptions of challenging news content*.

3 NEW RECOMMENDATION MODEL

The factors that are proposed for the diversity aware recommendation model are shown in Figure 1, and are described below in relation to: user diversity (e.g., personality), item diversity (e.g., re-ranking), and presentational strategies (e.g., item placement).

A combined study of both user- and item- diversity makes it possible to find solutions that address both user and algorithmic biases at once. By addressing the challenge of diversification from the angle of user perception, this diversity aware recommendation model builds on advances in the area of *presentational strategies* in recommender systems. An improved understanding of the factors that may influence the effectiveness of such a model make it possible to improve the positive impact of item diversification, and improve people’s acceptance of diverse news articles.

3.1 User Diversity

Users naturally have different interests, one of they key motivators for personalization algorithms such as those used in recommender systems. Previous studies have found that users vary in terms of the degree of diversification that is optimal for them, and that this can be deduced from their previous rating behavior (see e.g., [13, 21]). Users also vary in terms of personality traits that are fixed, such as the Big Five [12]. Our, and others’ work, has also found that recommendation algorithms may benefit from considering these traits [5, 16, 22]. One study investigated how people apply diversification for a sequence of book recommendations for a friend [22]. Others found that whether users were diversity-seeking or challenge-averse also influenced the perceptions of diversity in

news [16], a user trait that may be more transient. Beam found that demographic factors such as gender, level of internet skill and education affected the extent to which users reported to think at depth about news [3].

3.2 Item Diversity

The items that are recommended to a user, or considered as candidate recommendations, can also be selected in a way that they are different from each other. Several approaches to item diversification are suggested in the literature. Ziegler et al. (2005) proposed a topic diversification approach based on taxonomy-based dissimilarity [31]. As may be anticipated, using simple dissimilarity also impacted accuracy negatively. An alternate set of approaches which re-rank a list of top items was found to improve diversity without a great loss in accuracy (c.f., [1, 13]). Zhang et al. found that diversification in recommendations increased novelty and decreased “unserendipity” (similarity between items in a user’s history and new recommendations) [30].

These measures of diversity however suffer from a considerable limitation: they do not take into account whether they can be accepted or understood by a user. They also do not consider more subtle definitions of item diversity such as the the strength of sentiment, or preferences for certain styles of writing (stylometry). To understand which features influence users perceptions of item diversity, more exploratory studies, e.g., investigating users’ perceptions of diversity in active consumption environments are required (c.f., [24]).

3.3 Presentational Strategies

The diversity aware recommendation model will consider design choices such item placement, primacy and recency effects, transitions, and interaction mechanisms:

Item placement. The position of surprising or risky items, influences the perceived diversity of a list [9, 16]. A number of design decisions can be made around how items are grouped (or if they are spread out), where they are placed (e.g., beginning, middle, or end of a list), and how pair-wise distances between specific items are considered (e.g., transitions).

Similarity grouping refers to whether articles that are different from the rest of a recommended list, such as top-N, are more easily grouped together (as predicted by Gestalt principles) than when similarity within a list is more homogeneous. For example, Ge et al. studied the placement of diverse items, finding an effect on perceived recommendation list diversity [10]. Placing items that differ from the others in the middle in the list, and as a block (rather than spread out) were found to *reduce* perceived diversity.

Primacy and recency effects. Primacy and recency effects refer to the first and last positions in a list of recommendations: given that the first and last article in a list are normally the easiest to remember in recall tasks, algorithms which affect the ranking of position of articles in these positions are likely to influence user perceptions of sets of recommendations. Previous studies have found some effects of item placement at the beginning or end of a list. Sorting agreeable content first appears to decrease satisfaction rather than increasing it [16]. In contrast, placing the diverse items at the bottom of the list can increase the perceived diversity [9].

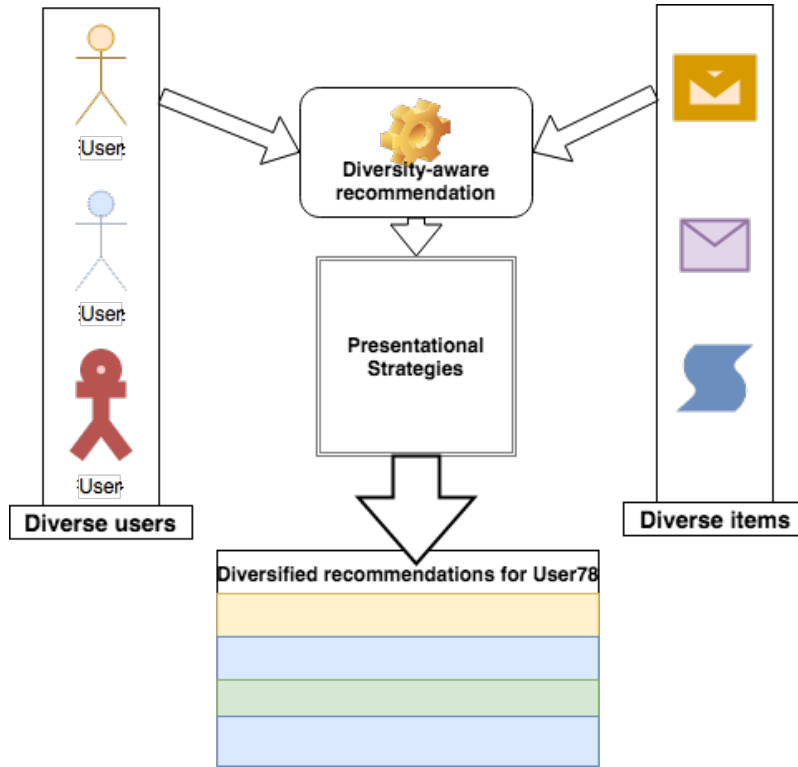


Figure 1: The proposed diversity aware recommendation model considers that both users and items are diverse. The resulting recommendation list consequently contains articles on a range of topics. The last item is on a highly relevant item represented in blue, but there are also yellow and green items in the list. These yellow and green items are relevant to User78 but are not necessarily the *most* similar to the user’s preferences.

Transitions. Transitions consider the size and types of gaps between pairs of items. Due to anchoring and other similar effects, the order of presentation matter. Intuitively, there are orderings that would be unsuitable for most users, like moving directly from a very sad news story to a very happy one, even if both stories are relevant to a given user.

Interaction Mechanisms. System designers can chose to introduce interaction mechanisms that help users manage diverse content. For example, explicit explanation mechanisms, such as textual explanations for surprising items, may help users understand the choice of specific item. Systems can also include implicit interaction mechanisms, such as linking the recommendation list to a visual interface to support exploration (c.f., [26]). The interaction can also be designed to help users both discover and explore their blindspots.

4 CONCLUSION AND OUTLOOK

Building on previous research in information presentation, this paper suggests approaching diversification from a more user-centered approach than has been previously considered. To address filter bubbles, we consider the problem from *both* a computational and user-centered point of view. This is the first attempt to create a diversity-aware recommendation framework that considers how presentational strategies can help aid the diversification of content.

This work allows us to better understand how to maximally increase the diversity of content a user is exposed to, while maintaining user satisfaction.

By considering both user and item diversity this approach is a unique and valuable contribution toward addressing the issue of over tailoring, leading to more balanced news consumption. By doing so, this *diversity aware recommendation model* enables us to address the challenges of both user and algorithmic biases, which often conspire to the creation of filter bubbles.

In line with this vision, first steps have been taken to study how different presentational strategies influence perceptions of diversity, a.o., studying the effects of different kinds of transitions between diverse items (under preparation), and users’ expectations, and perceptions, of diversity in playlists [24]. We will also continue to build on our previous work on explanation interfaces that used weak ties to support content discovery [14, 23], to study the role of item positions in relation to perceptions of diversity. By defining diversity in a way that is understandable and acceptable to users, it becomes possible to move research on explanation-aware recommendation to the next level: how we present diverse items in recommender systems can help users not only to understand the recommendations, but also themselves and their own biases. In doing so, it may be possible to maximize the amount of diverse content that users are exposed to, without damaging system reputation.

REFERENCES

- [1] Gediminas Adomavicius and YoungOk Kwon. 2011. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *IEEE Transactions on Knowledge and Data Engineering* 24 (2011), 896–911.
- [2] Eytan Bakshy, Solomon Messing, and Lada A. Adamic. 2015. Exposure to ideologically diverse news and opinion on Facebook. *Science* 348 (2015), 1130–1132.
- [3] Michael A Beam. 2014. Automating the news: How personalized news recommender system design choices impact news reception. *Communication Research* 41, 8 (2014), 1019–1041.
- [4] Derek Bridge and John Paul Kelly. 2006. Ways of Computing Diverse Collaborative Recommendations. In *Adaptive Hypermedia and Adaptive Web-based Systems*. 41–50.
- [5] Li Chen, Wen Wu, and Liang He. 2016. Personality and Recommendation Diversity. In *Emotions and Personality in Personalized Services*. Springer, 201–225.
- [6] Michael D Conover, Bruno Gonçalves, Alessandro Flammini, and Filippo Menczer. 2012. Partisan asymmetries in online political activity. *EPJ Data Science* 1, 1 (2012), 6.
- [7] Bruce Ferwerda, Mark P Graus, Andreu Vall, Marko Tkalcić, and Markus Schedl. 2017. How item discovery enabled by diversity leads to increased recommendation list attractiveness. In *Proceedings of the Symposium on Applied Computing*. ACM, 1693–1696.
- [8] DJ Flynn, Brendan Nyhan, and Jason Reifler. 2016. The Nature and Origins of Misperceptions: Understanding False and Unsupported Beliefs about Politics. *Advances in Pol. Psych* (2016).
- [9] Mouzhi Ge, Carla Delgado-Battenfield, and Dietmar Jannach. 2010. Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. In *Recommender Systems*.
- [10] Mouzhi Ge, Dietmar Jannach, Fatih Gedikli, and Martin Hepp. 2012. Effects of the Placement of Diverse Items in Recommendation Lists. In *ICEIS*. 201–208.
- [11] Eduardo Graells-Garrido, Mounia Lalmas, and Ricardo Baeza-Yates. 2016. Data Portraits and Intermediary Topics: Encouraging Exploration of Politically Diverse Profiles. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 228–240.
- [12] Oliver P John and Sanjay Srivastava. 1999. The Big Five trait taxonomy: History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research* 2, 1999 (1999), 102–138.
- [13] Michael Jugovac, Dietmar Jannach, and Lukas Lerche. 2017. Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems with Applications* 81 (2017), 321–331.
- [14] Byungkyu Kang, Nava Tintarev, Tobias Höllerer, and John O'Donovan. 2016. What am I not Seeing? An Interactive Approach to Social Content Discovery in Microblogs. In *Social Informatics*, Emma Spiro and Yong-Yeol Ahn (Eds.). Springer International Publishing, 279–294.
- [15] M. Mandl and A. Felfernig. 2012. Improving the Performance of Unit Critiquing. In *UMAP 2012*. Montreal, Canada, 176–187.
- [16] Sean A. Munson and Paul Resnick. 2010. Presenting Diverse Political Opinions: How and How Much. In *CHI*.
- [17] Sayooran Nagulendra and Julita Vassileva. 2014. Understanding and controlling the filter bubble through interactive visualization: a user study. In *Proceedings of the 25th ACM conference on Hypertext and social media*. ACM, 107–115.
- [18] Nic Newman, David AL Levy, and Rasmus Kleis Nielsen. 2015. Reuters Institute Digital News Report 2015. (2015).
- [19] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*. ACM, 677–686.
- [20] Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you*. Penguin Books.
- [21] Yue Shi, Xiaoxue Zhao, Jun Wang, Martha Larson, and Alan Hanjalic. 2012. Adaptive diversification of recommendation results via latent factor portfolio. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 175–184.
- [22] Nava Tintarev, Matt Dennis, and Judith Masthoff. 2013. Adapting Recommendation Diversity to Openness to Experience: A Study of Human Behaviour. In *User Modeling, Adaptation, and Personalization*. Springer, 190–202.
- [23] Nava Tintarev, Byungkyu Kang, T. Höllerer, and John O'Donovan. 2015. Inspection Mechanisms for Community-based Content Discovery in Microblogs. In *Joint Workshop on Interfaces and Human Decision Making for Recommender Systems in conjunction with Recsys*.
- [24] Nava Tintarev, Christoph Lofi, and Cynthia C. S. Liem. 2017. Sequences of Diverse Song Recommendations: An exploratory study in a commercial system. In *User Modeling, Adaptation, and Personalization*. 391–392.
- [25] Nava Tintarev and Judith Masthoff. 2012. Evaluating the effectiveness of explanations for recommender systems: Methodological issues and empirical studies on the impact of personalization. *User Modeling and User-Adapted Interaction* 22 (2012), 399–439.
- [26] Chun-Hua Tsai and Peter Brusilovsky. 2017. Enhancing Recommendation Diversity Through a Dual Recommendation Interface. In *Workshop on Interfaces and Human Decision Making for Recommender Systems*.
- [27] Chun-Hua Tsai and Peter Brusilovsky. 2017. Providing Control and Transparency in a Social Recommender System for Academic Conferences. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 313–317.
- [28] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welp. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM* 10, 1 (2010), 178–185.
- [29] Saul Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 109–116.
- [30] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. 2012. Auralist: introducing serendipity into music recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 13–22.
- [31] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. 2005. Improving Recommendation Lists Through Topic Diversification. In *World Wide Web (WWW)*.

Academic performance prediction in a gender-imbalanced environment

Piotr Sapiezynski
Northeastern University

Valentin Kassarnig
Graz University of Technology

Christo Wilson
Northeastern University

Sune Lehmann
Technical University of Denmark

Alan Mislove
Northeastern University

ABSTRACT

Individual characteristics and informal social processes are among the factors that contribute to a student's performance in an academic context. Universities can leverage this knowledge to limit drop-out rates and increase performance through interventions targeting at-risk students. Data-driven recommendation systems have been proposed to identify such students for early interventions. However, as we show in this paper, it is possible to identify certain groups of students whose performance is best predicted using indicators that differ from those predictive for the majority. Naïve approaches that do not account for this fact might favor the majority class and lead to disparate mistreatment in the case of minorities. In this paper we investigate the low academic performance predictors of female and male participants of the Copenhagen Networks Study. We find that social indicators (e.g. mean grade point average of peers or fraction of low-performing peers) predict low-performance of male participants more accurately than they do for female participants, and that this situation is reversed for individual behaviors. Because of the gender imbalance among the participants, optimal gender-oblivious models detect low-performing male students with higher accuracy than low-performing female students. We review the existing approaches to addressing the disparate mistreatment problem and propose our own method that outperforms the alternatives on the dataset in question.

ACM Reference format:

Piotr Sapiezynski, Valentin Kassarnig, Christo Wilson, Sune Lehmann, and Alan Mislove. 2017. Academic performance prediction in a gender-imbalanced environment. In *Proceedings of FATREC Workshop on Responsible Recommendation at ACM RecSys, Como, Italy, August 2017 (FATREC'17)*, 4 pages.
<https://doi.org/10.18122/B20Q5R>

1 INTRODUCTION

One of the central driving forces behind the adoption of algorithmic decision-making is the goal of eliminating biases from the decision process. However, it has recently been shown that these algorithms can have the opposite effect, possibly as a consequence of how the data is mined [2]. Algorithmic biases have been demonstrated in the systems that make decisions (or aid the human decision making process) in areas as diverse as loans [10], parole [12], hiring [10], and policing [15].

A growing body of fairness research emphasizes a range of problems with black box algorithms. There exist multiple definitions of fairness, some of which have been shown to be mutually exclusive [9]. The discussion is especially heated around *disparate mistreatment*: a situation in which error rates in a decision making process are not balanced between representatives of a particular characteristic (e.g. gender or race). Angwin et. al. [12] argued that the system judges use as an assistant in their parole decisions is more likely to wrongly imprison blacks than whites. The article provoked a series of responses, which argued that the system was indeed fair, but according to a different definition of fairness [6, 8]. The notion of disparate mistreatment was formalized by Zafar et al. in a recent article which also introduces an approach of solving the problem through constrained training of the classifier [23].

Independently of the research on fairness, there is increasing interest in data-driven predictions of academic performance and intervention recommendations. For example Balfanz, et al. [1] proposed a system based on school records that recommends targeted interventions to activate students at high risk of dropping out from high school. More recently, Wang et al. [21] showed that the academic performance can also be predicted from behavioral data collected using smartphones. In a student population we studied recently, social indicators proved to be more predictive of academic performance than the behavior or characteristics of the individual [14]. These social factors (including mean grade point average of peers and the fraction of low-performing peers) were more highly correlated with an individual performance than, for example, class attendance. In this paper, we ask whether these findings hold equally for men and women in the dataset. Further, we ask whether a model built on these features works equally well for the two sexes. Finally, we review the existing methods of avoiding disparate mistreatment and propose a novel approach, based on constrained forward feature selection. Instead of optimizing the classifier for best overall performance, we constrain the training process by progressively adding features so that the model maintains comparable performance for all groups of the protected feature (i.e. for men and women). While this simple approach might not work on datasets where balanced features are absent, it does outperform other methods on our dataset. Of course, while our method can accurately identify low-performing male and female students, recommending particular interventions lies beyond the scope of this study.

This article may be copied, reproduced, and shared under the terms of the Creative Commons Attribution-ShareAlike license (CC BY-SA 4.0).

FATREC'17, August 2017, Como, Italy

© 2017 Copyright held by the owner/author(s).

DOI: 10.18122/B20Q5R

Table 1: Summary statistics of the dataset. There is no statistically significant difference between performance among men and women in the study ($p_{val} = 0.65$ in Kolomogorov-Smirnov test)

	Performance			Total
	Low	Medium	High	
Male	142	141	137	420
Female	38	39	43	120
Total	180	180	180	540

2 METHODS

2.1 Data

The data used in this paper was collected as part of the Copenhagen Networks Study (CNS), a large scale computational social science study designed to measure human interactions and mobility with high resolution [20]. The approximately 800 participants of the study were freshmen and sophomores at the Technical University of Denmark. After responding to an online questionnaire on psychological and health indicators, they were equipped with an instrumented smartphone (Google Nexus 4) that—with their consent—tracked their location, proximity to other participants, and communication instances (metadata of short messages and calls, without the content). Finally, the vast majority of the participants (717 out of 839) opted in to share their Facebook data as well, which was acquired using Facebook API. The data collection campaign lasted two years. In this study we focus on participants who interacted with at least three other subjects through phone calls, short messages, face to face, and on Facebook. There are 420 men and 120 women in the dataset, and this gender imbalance corresponds to the imbalance in the overall student population. We divide the students into three equally-sized groups based on their GPA after two years. Table 1 presents summary statistics.

We derive a number of variables in the following feature categories:

Individual behaviors. *Class attendance* is computed from location data combined with class schedule using the method we previously described [13]; it corresponds to the fraction of lectures and exercises a student attended within the courses they signed up for. *Facebook activity* score is defined as the mean number of status updates a student posted in a week during the duration of the observation.

Individual characteristics. This dataset was obtained through an online questionnaire and includes: The Big Five [11] (*neuroticism*, *openness*, *conscientiousness*, *extraversion*, *agreeableness*), Rotter’s *Locus of Control* [18], *stress* [4], *self-esteem* [17], *satisfaction with life* [5], PANAS (*positive* and *negative*) [22], *loneliness* [19], *depression* [3], and narcissism (*rivalry*, *admiration*, *overall*) [7].

Network characteristics. *Degree centrality* measures, one for each of four interactions networks: in physical space (person-to-person proximity measured using Bluetooth), calls and short message exchanges, and Facebook interactions.

Peer performance. Knowing the underlying social networks (proximity, phone communication, and Facebook) as well as

the grades of each participant, we computed the *mean GPA* of each persons’ peers, as well as *fraction of low/high-performers* (two features for each interaction network).

2.2 Classifier training

In each problem, we train a common classifier, oblivious to gender. We use k -fold cross-validation with $k = 3$ (due to the low number of female samples in the dataset we maintained a small k to avoid folds with no women). In each test fold, we calculate the performance on (a) all test samples, (b) only male samples, and (c) only female samples, and report these in figures. As we showed in our previous work [14], Linear Discriminant Analysis (LDA) is the machine learning approach that achieves the highest results with the dataset (compared against logistic regression, random forest, and SVC). We tune hyper-parameters through grid search cross-validation separately for each feature-set.

3 RESULTS

3.1 Detecting low-performing students

We divide students into three equally sized groups based on their grade point average (GPA): low-, mid-, and high-performing students. In this article we focus on identifying low performing students. Hence, we rephrase the problem as a binary classification task, where the target class are the low-performers, consisted with identifying students to intervene. We then use four fine-tuned LDA models to predict student performance each based on a different feature-set: individual characteristics, individual behaviors, network centrality, and peer performance. We then combine first two categories and train the ‘individual’ model; we combine the third and fourth sets and train the ‘network’ model. We then combine all features into a ‘combined’ model.

As shown in Figure 1, peer-performance is a good predictor of low performance amongst men, but the signal is weaker for female students. Combining the individual and network features into a common model results in a gap in predictive performance between men and women ($AUC\ ROC = 0.84$ and 0.67 , respectively). To better illustrate this effect, we investigate example cumulative distributions of social and individual features among the genders with respect to performance, see Figure 2.

3.2 Fair predictions through feature selection

Now we build a model which maximizes a prediction performance metric in the low-performers’ detection problem, while constraining the difference of performance between genders. We adapt a forward feature selection strategy: we start by selecting the feature that has the highest predictive power for the entire population while satisfying the requirement given in Eq. 1:

$$\frac{|P_m - P_w|}{P_{total}} \leq \epsilon, \quad (1)$$

where ϵ is a parameter controlling how much inter-gender difference we are willing to allow, and P is the selected performance metric, for example area under receiver characteristic curve ($AUC\ ROC$), or Matthew’s Correlation Coefficient (MCC). We then add more features, one by one, in a way that the new model has increasing P score and satisfies the requirement from Eq. 1.

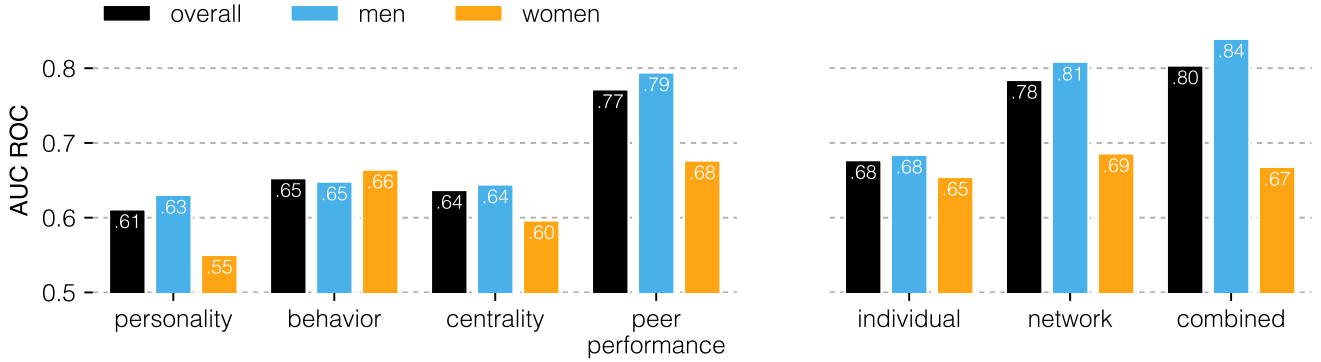


Figure 1: Low-performers' detection. Peer-performance is an efficient predictor of low performance amongst men, but the signal is much weaker for female students. Note, that the *AUC ROC* of a random classifier would be equal to 0.5, so all feature categories provide signal related to low academic performance.

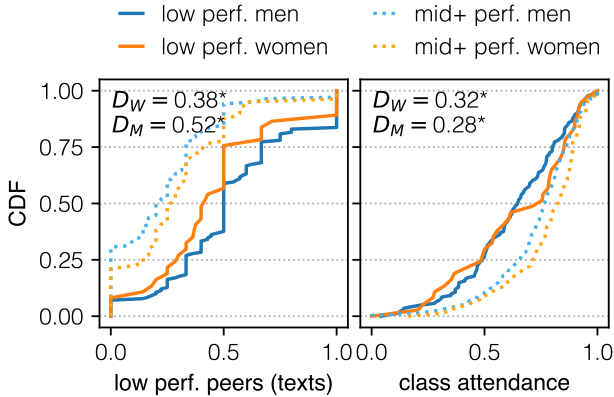


Figure 2: We use the Kolmogorov-Smirnov test on cumulative distribution functions (CDF) of two features (*fraction of low-performing peers* in the text network, and *class attendance*) to measure how dissimilar low-performing students of each gender are from the high performers. We find larger differences for men than women in the peer performance feature. However, the difference is larger for women in the individual behavior feature. Annotated are the results of K-S test, marked with the (*) symbol wherever significant with $p_{val} < 0.05$.

Figure 3 shows the results of training such fair classifiers. It emphasizes the trade-off between overall performance and fairness: the bigger the allowed difference between genders, the higher the overall performance. Typically, in binary classification tasks *AUC ROC* is used to measure the performance of the classifier. In this case, however, using *AUC ROC* might be misleading: it summarizes the performance of a classifier at all thresholds, but a classifier put to use would have to operate at a chosen threshold. Even if *AUC ROC* scores are balanced, the classifier at a particular threshold might still suffer from the disparate mistreatment problem. Therefore, we

perform the constrained forward feature selection using Matthew's correlation coefficient [16]. It quantifies the performance at a threshold and—contrary to the popularly used F_1 score—penalizes the classifier for classifying all samples as the target class (such a classifier on this dataset has $MCC = 0$ and $F_1 = 0.5$). We define MCC in Eq 2.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (2)$$

3.3 Alternative approaches

Figure 4 compares the results achieved through constrained forward feature selection (CFFS), the method proposed by Zafar et al. [23], re-balancing the dataset, as well as training separate models for men and women. Because of too few female subjects in the data, training separate models results in severe penalty on performance of the female-only model. Re-balancing the dataset as well as the approach proposed by Zafar et al. [23] achieve better results. Constrained forward feature selection achieves high and nearly equal MCC for both genders.

4 DISCUSSION

In this work we showed that empirical data can be more predictive for a one group of subjects than other groups, and the problem might go unnoticed unless specifically investigated. The situation we described is not simply the case of imbalance, as re-balancing the data does not solve the issue. Instead, we found that fair learning can be achieved by only learning on selected features. The solution is not generalizable to all datasets—depending on the problem, there might be no features that perform similarly well for representants of all classes among the protected feature. We tested our approach on other datasets. It fails, for example, to solve the disparate mistreatment problem in the COMPAS dataset [12], where all predictive features achieve higher performance for one of the races. Therefore, rather than recommending our approach for use in all scenarios, we limit our conclusion to emphasizing the need for considering the diversity of users in machine learning systems.

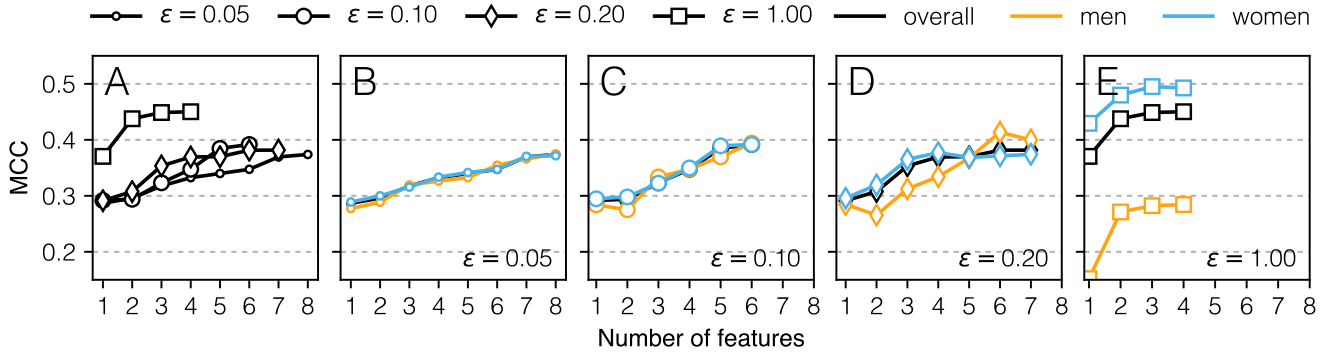


Figure 3: Learning fair classifiers. In each step we extend the model with a feature to maximize the overall performance of the classifier while maintaining the maximum disparity ϵ between genders. $\epsilon = 1$ means there is no constraint on parity. Note, that a constrained classifier has a higher performance for the underrepresented class than the unconstrained classifier. Note that for a random classifier $MCC = 0$. The selection process stops when no more features can be added to improve performance while maintaining performance parity, hence a possible difference in the number of features used depending on ϵ .

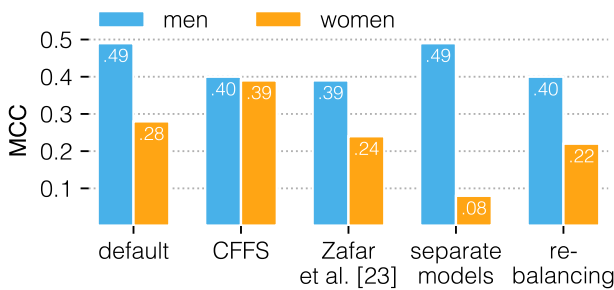


Figure 4: Alternative approaches to learning fair classifiers. On the dataset in question, the constrained forward feature selection (CFFS) method outperforms other approaches.

REFERENCES

- [1] Robert Balfanz, Liza Herzog, and Douglas J. Mac Iver. 2007. Preventing Student Disengagement and Keeping Students on the Graduation Path in Urban Middle-Grades Schools: Early Identification and Effective Interventions. *Educational Psychologist* 42, 4 (2007), 223–235. <https://doi.org/10.1080/00461520701621079> arXiv:<http://dx.doi.org/10.1080/00461520701621079>
- [2] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *California Law Review* 104 (2016), 671–732. <https://ssrn.com/abstract=2477899>
- [3] P Bech, N-A Rasmussen, L Raabæk Olsen, V Noerholm, and W Abildgaard. 2001. The sensitivity and specificity of the Major Depression Inventory, using the Present State Examination as the index of diagnostic validity. *Journal of affective disorders* 66, 2 (2001), 159–164.
- [4] Sheldon Cohen, Tom Kamarck, and Robin Mermelstein. 1983. A global measure of perceived stress. *Journal of health and social behavior* (1983), 385–396.
- [5] Ed Diener, Robert A. Emmons, Randy J. Larsen, and Sharon Griffin. 1985. The satisfaction with life scale. *Journal of Personality Assessment* 49, 1 (1985), 71–75.
- [6] William Dieterich, Christina Mendoza, and Tim Brennan. 2016. COMPAS risk scales: Demonstrating accuracy equity and predictive parity. Technical Report. Technical report, Northpointe, July 2016. <http://www.northpointeinc.com/northpointe-analysis>.
- [7] Robert A Emmons. 1984. Factor analysis and construct validity of the narcissistic personality inventory. *Journal of personality assessment* 48, 3 (1984), 291–300.
- [8] Anthony W Flores, Kristin Bechtel, and Christopher T Lowenkamp. 2016. False Positives, False Negatives, and False Analyses: A Rejoinder to Machine Bias: There's Software Used across the Country to Predict Future Criminals. And It's Biased against Blacks. *Fed. Probation* 80 (2016), 38.
- [9] Sorelle A. Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2016. On the (im)possibility of fairness. *CoRR* abs/1609.07236 (2016). <http://arxiv.org/abs/1609.07236>
- [10] Anikó Hannák, Claudia Wagner, David Garcia, Alan Mislove, Markus Strohmaier, and Christo Wilson. 2017. Bias in Online Freelance Marketplaces: Evidence from TaskRabbit and Fiverr. In *CSCW*. 1914–1933.
- [11] Oliver P John and Sanjay Srivastava. 1999. The Big Five trait taxonomy: History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research* 2, 1999 (1999), 102–138.
- [12] Angwin Julia, Larson Jeff, Mattu Surya, and Lauren Kirchner. 2016. Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And it's Biased Against Blacks. *ProPublica* (2016). <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>
- [13] Valentin Kassarnig, Andreas Bjerre-Nielsen, Enys Mones, Sune Lehmann, and David Dreyer Lassen. 2017. Class attendance, peer similarity, and academic performance in a large field study. *CoRR* abs/1702.01262 (2017). <http://arxiv.org/abs/1702.01262>
- [14] Valentin Kassarnig, Andreas Bjerre-Nielsen, Enys Mones, Piotr Sapiezynski, Sune Lehmann, and David Dreyer Lassen. 2017. Academic Performance and Behavioral Patterns. (2017). Under review.
- [15] Kristian Lum and William Isaac. 2016. To predict and serve? *Significance* 13, 5 (2016), 14–19. <https://doi.org/10.1111/j.1740-9713.2016.00960.x>
- [16] Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405, 2 (1975), 442–451.
- [17] Morris Rosenberg. 1965. *Society and the adolescent self-image*. Princeton university press Princeton, NJ.
- [18] Julian B Rotter. 1966. Generalized expectancies for internal versus external control of reinforcement. *Psychological monographs: General and applied* 80, 1 (1966), 1.
- [19] Daniel W Russell. 1996. UCLA Loneliness Scale (Version 3): Reliability, validity, and factor structure. *Journal of Personality Assessment* 66, 1 (1996), 20–40.
- [20] Arkadiusz Stopczynski, Vedran Sekara, Piotr Sapiezynski, Andrea Cuttone, Mette My Madsen, Jakob Eg Larsen, and Sune Lehmann. 2014. Measuring large-scale social networks with high resolution. *PloS one* 9, 4 (2014), e95978.
- [21] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. 2014. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 3–14.
- [22] David Watson, Lee A Clark, and Auke Tellegen. 1988. Development and validation of brief measures of positive and negative affect: the PANAS scales. *Journal of personality and social psychology* 54, 6 (1988), 1063.
- [23] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1171–1180.

Considerations on Recommendation Independence for a Find-Good-Items Task

Toshihiro Kamishima and Shotaro Akaho

National Institute of Advanced Industrial Science and Technology (AIST)
AIST Tsukuba Central 2, Umezono 1-1-1, Tsukuba, Ibaraki, Japan 305-8568
mail@kamishima.net(<http://www.kamishima.net/>), s.akaho@aist.go.jp

ABSTRACT

This paper examines the notion of recommendation independence, which is a constraint that a recommendation result is independent from specific information. This constraint is useful in ensuring adherence to laws and regulations, fair treatment of content providers, and exclusion of unwanted information. For example, to make a job-matching recommendation socially fair, the matching should be independent of socially sensitive information, such as gender or race. We previously developed several recommenders satisfying recommendation independence, but these were all designed for a predicting-ratings task, whose goal is to predict a score that a user would rate. We here focus on another find-good-items task, which aims to find some items that a user would prefer. In this task, scores representing the degree of preference to items are first predicted, and some items having the largest scores are displayed in the form of a ranked list. We developed a preliminary algorithm for this task through a naive approach, enhancing independence between a preference score and sensitive information. We empirically show that although this algorithm can enhance independence of a preference score, it is not fit for the purpose of enhancing independence in terms of a ranked list. This result indicates the need for inventing a notion of independence that is suitable for use with a ranked list and that is applicable for completing a find-good-items task.

CCS CONCEPTS

• Information systems → Recommender systems;

KEYWORDS

fairness, recommendation independence, matrix factorization

ACM Reference format:

Toshihiro Kamishima and Shotaro Akaho. 2017. Considerations on Recommendation Independence for a Find-Good-Items Task. In *Proceedings of Workshop on Responsible Recommendation, Como, Italy, August 2017*, 6 pages. <https://doi.org/10.18122/B2871W>

1 INTRODUCTION

Recommender systems and other personalization technologies, which help to search for items or information predicted to be useful to a user, have become indispensable tools in support of decision-making. To avoid unfairness or bias in the decisions supported by recommender systems, the influence of specific information should be excluded from the prediction process of recommendation.

In other words, independence between recommendation results and specific information should be maintained in the following situations. First, recommendation services must be managed in adherence to laws and regulations. Sweeny presented an example of dubious advertisement placement that appeared to exhibit racial discrimination [21]. In this case, the selection of personalized advertisements should be rendered independent of racial information. Another concern is the fair treatment of information providers. The Federal Trade Commission has been investigating Google to determine whether the search engine ranks its own services higher than those of competitors [3]. In this case, no deliberate manipulation was found. However, an algorithm that can explicitly exclude information about whether content providers are competitors would be helpful for alleviating users' doubts as well as competitors' doubts about unfair manipulations. Finally, recommendation independence is helpful for excluding the influence of unwanted information. Popularity bias, which is the tendency for frequently consumed items to be recommended more frequently [2], is a well-known drawback of recommenders. If information about popularity could be excluded, users could acquire information free from unwanted popularity bias. In summary, excluding the influence of specific information is helpful for the following purposes: adherence to laws and regulations, fair treatment of content providers, and exclusion of unwanted information.

To fulfill the need for excluding the influence of specific information, we formalized a notion of recommendation independence and developed algorithms to enhance it. For this purpose, we exploited a technique developed for fairness-aware data mining [5, 17], whose goal is to analyze data while taking into account potential issues of fairness. Following the notions proposed in the previous studies, we formally define recommendation independence as statistical independence between a recommendation result and specified information. In addition, we developed an independence-enhanced recommender system (IERS) that could satisfy a constraint of recommendation independence [9]. This IERS is also technically challenging and non-trivial, because while there are many techniques for incorporating new types of information, there are very few trials to exclude unwanted information. We developed two approaches for enhancing recommendation independence. One was a regularization approach, which adopted an objective function with a constraint term for imposing recommendation independence [9, 11, 12]. The other was a model-based approach, which adopted a generative model in which ratings and sensitive features were independent [13].

However, all our previous methods targeted a predicting-ratings task, predicting a score of items that a user would rate, although there are other types of recommendation tasks. One such task is a

This article may be copied, reproduced, and shared under the terms of the Creative Commons Attribution-ShareAlike license (CC BY-ND 4.0).

Workshop on Responsible Recommendation, August 2017, Como, Italy

© 2017 Copyright held by the owner/author(s).

DOI: 10.18122/B2871W

find-good-items task, whose goal is to find some items that a user would prefer [4, 8]. To complete this type of task, a system predicts preference scores, which quantify how strongly a target user prefers items, for every candidate item. These items are then displayed to a target user in the form of a ranked list sorted according to the predicted scores.

In this paper, we investigate recommendation independence for this find-good-items task. In the case of a predicting-ratings task, we enhanced independence between a predicted rating and a sensitive feature. However, in the find-good-items case, the notion of independence between a ranked list and a sensitive feature is unclear. We therefore examine a naive approach, treating independence between a preference score used for ranking items and a sensitive feature. We develop a preliminary recommendation method to enhance this type of independence by a regularization approach. By applying this method, we empirically inspect the independence from a preference score or a ranked list.

Our contributions can be summarized as follows.

- We develop a preliminary recommendation method for a find-good-items task through an approach of enhancing the independence of a preference score from a sensitive feature.
- We empirically show that the independence of a preference score could be enhanced without sacrificing prediction accuracy.
- However, our experimental results reveal that the determination as to whether items are relevant is not always independent from a sensitive feature.

These results lead to the conclusion that we must develop a new notion of recommendation independence fitting for a find-good-items task.

This paper is organized as follows. In section 2, we formalize the concept of recommendation independence and an IERS task. We show our new method for enhancing recommendation independence in section 3. Our experimental results are shown in section 4. Related work is discussed in section 5, and section 6 concludes our paper.

2 RECOMMENDATION INDEPENDENCE

This section describes a formal definition of recommendation independence and an independence-enhanced recommendation task.

2.1 Definition

To formalize recommendation independence, we need to specify a *sensitive feature*, using the terminology from studies in the fairness-aware data mining literature [5, 17]. We can then attempt to maintain recommendation independence from this sensitive feature, denoted by S . In Sweeney’s example of advertisement placement described in section 1, racial information corresponds to a sensitive feature. R represents a recommendation result, which is the degree of relevance to a user’s preference used for sorting candidate items in this paper. Based on information theory, the statement “information about a sensitive feature is excluded from the prediction process of the recommendation” describes the condition in which mutual information between R and S is zero. This condition is equivalent to statistical independence between R and S , i.e., $\Pr[R] = \Pr[R|S]$.

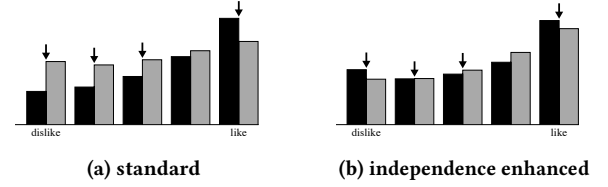


Figure 1: Distributions of the predicted preference scores for each sensitive value

To illustrate the effect of enhancing recommendation independence, we show distributions of predicted preference scores in Figure 1. The charts in this figure show experimental results for ML1M-Year data using an independence parameter, $\eta=10$. The details of the experimental conditions will be shown in section 4. Black and gray bars show the distributions of predicted scores for older and newer movies, respectively. In Figure 1(a), scores are predicted by a standard recommendation algorithm, and older movies are highly rated (see the big gaps between two bars indicated by arrowheads). When recommendation independence is enhanced as in Figure 1(b), the distributions of scores for older and newer movies become much closer (the large gaps are lessened); that is to say, the predicted ratings are less affected by a sensitive feature.

We here note why a sensitive feature must be specified in the definition of recommendation independence. In brief, a sensitive feature must be selected because it is intrinsically impossible to personalize recommendation results if the results are independent of all features. This is due to the *ugly duckling theorem*, which asserts the impossibility of classification without weighing certain features as more important than others [22]. Because recommendation is considered as a task for classifying whether or not items are preferred, certain features inevitably must be weighed. Consequently, it is impossible to enhance independence from all features equally. In the RecSys2011 panel [18], a panelist also pointed out that no information is neutral, and thus individuals are always influenced by information biased in some sense.

2.2 Task Formalization

We formalize a recommendation task whose independence is enhanced. We previously targeted a predicting-ratings recommendation task, which predicted a ratings of items given by a user [4]. In this paper, we concentrate on a find-good-items task, whose goal is to find some items that a user would prefer. $X \in \{1, \dots, n\}$ and $Y \in \{1, \dots, m\}$ denote random variables for the user and item, respectively. x and y are instances of X and Y , respectively. We here assume that users explicitly show their preference for items. In a predicting-ratings case, R denotes a random variable that expresses the rating of an item. To fit our previous algorithms for use with a find-good-items task, we make R denote whether an item is relevant or irrelevant to a user. When presenting an item x to a user y , $R=1$ if the item is relevant to the user; otherwise $R=0$. To complete an IERS task, we additionally need a sensitive feature, S , from which independence will be enhanced. The domain of S is currently restricted to a binary type, $\{0, 1\}$, for simplicity.

One training datum consists of a user, x , an item, y , a sensitive value, s (an instance of S), and relevance information, r (an instance of R). A training dataset is the set of N data, $\mathcal{D} =$

$\{(x_i, y_i, s_i, r_i)\}$, $i = 1, \dots, N$. We define $\mathcal{D}^{(s)}$ as a subset consisting of all data in \mathcal{D} whose sensitive value is s . Given a new datum, (x, y, s) , a preference function, $\hat{r}(x, y, s)$, predicts a preference score of the item y for the user x . The aim of an IERS task is to learn this preference function to predict a preference score, indicating the degree of relevance, from a given training dataset under the constraint of recommendation independence. The prediction accuracy generally decreases when an independence constraint is satisfied, due to the loss of usable information. Therefore, it is desirable to satisfy the constraint while sacrificing as little accuracy possible as possible.

3 AN IERS FOR A FIND-GOOD-ITEMS TASK

This section shows a logistic probabilistic-matrix-factorization model. We then introduce an independence-enhanced variant of this model by using a technique in [11].

3.1 A Logistic Matrix Factorization Model

We first introduce a logistic matrix factorization model for a find-good-items task. In our previous algorithms for a predicting-ratings task, we used a probabilistic matrix factorization (PMF) model [14]. Unlike the predicting-ratings case, a target preference, R , can take a value of only 0 or 1 in a find-good-items case. We hence apply a sigmoid function, which is a technique used in [19], and obtain a preference function:

$$\hat{r}(x, y) = \text{sig}(\mu + b_x + c_y + \mathbf{p}_x^\top \mathbf{q}_y), \quad (1)$$

where μ , b_x , and c_y are global, per-user, and per-item bias parameters, respectively, and \mathbf{p}_x and \mathbf{q}_y are K -dimensional parameter vectors, which represent the cross effects between users and items. $\text{sig}(a)$ denotes a sigmoid function, $1/(1 + \exp(-a))$. We call this a logistic probabilistic matrix factorization (logistic PMF) model.

3.2 An Independence-Enhanced Logistic PMF Model

We then show an independence-enhanced variant of a logistic PMF model. We use a regularization approach, which was originally developed for a fairness-aware classification task [10]. In this approach, we add an independence term to impose a constraint of recommendation independence. We advocated a simple independence term that was designed to match two means of predicted ratings for $\mathcal{D}^{(0)}$ and $\mathcal{D}^{(1)}$ [11].

We first modified a logistic PMF model (1) so that it depended on a sensitive value. For each value of $s \in \{0, 1\}$, we prepared parameter sets, $\mu^{(s)}$, $b_x^{(s)}$, $c_y^{(s)}$, $\mathbf{p}_x^{(s)}$, and $\mathbf{q}_y^{(s)}$. One of the parameter sets was chosen according to the sensitive value, and we obtained the preference function, as follows:

$$\hat{r}(x, y, s) = \text{sig}(\mu^{(s)} + b_x^{(s)} + c_y^{(s)} + \mathbf{p}_x^{(s)\top} \mathbf{q}_y^{(s)}). \quad (2)$$

We fit this model so as to minimize the following cross-entropy loss, instead of a squared loss used in a predicting-ratings case, because

a domain of R is restricted to 0 or 1:

$$\text{loss}(\mathcal{D}) = \sum_{(x_i, y_i, r_i, s_i) \in \mathcal{D}} - \left(r_i \log \hat{r}(x_i, y_i, s_i) + (1 - r_i) \log(1 - \hat{r}(x_i, y_i, s_i)) \right). \quad (3)$$

Next, we introduce an independence term to impose recommendation independence. This term quantifies the expected degree of independence between a predicted preference and a sensitive feature, with larger values indicating higher levels of independence. The independence term proposed in [11] was designed so as to make the two distributions $\Pr[R|S=0]$ and $\Pr[R|S=1]$ similar, because R and S become statistically independent if $\Pr[R|S=0] = \Pr[R|S=1]$. We thus used a squared norm between the means of these distributions, and the independence term became

$$\text{indep}(R, S) = - \left(\frac{\mathbb{S}^{(0)}}{|\mathcal{D}^{(0)}|} - \frac{\mathbb{S}^{(1)}}{|\mathcal{D}^{(1)}|} \right)^2, \quad (4)$$

where $\mathbb{S}^{(s)}$ is the sum of predicted preferences over the set $\mathcal{D}^{(s)}$,

$$\mathbb{S}^{(s)} = \sum_{(x_i, y_i, s_i) \in \mathcal{D}^{(s)}} \hat{r}(x_i, y_i, s_i). \quad (5)$$

Finally, we defined an objective function used in the regularization approach. The objective function is the sum of a loss term (3), an independence term (4), and an L_2 regularizer:

$$\text{loss}(\mathcal{D}) - \eta \text{indep}(R, S) + \lambda \text{reg}(\Theta), \quad (6)$$

where $\eta > 0$ is an independence parameter to balance the loss and independence, $\lambda > 0$ is a regularization parameter, and $\text{reg}(\Theta)$ is an L_2 regularizer to avoid over-fitting. By minimizing this objective, the parameters of models can be estimated so that the learned prediction function makes accurate predictions and satisfies the constraint of recommendation independence. Once the parameters of a model are estimated, preference scores for new data can be predicted by a prediction function (2).

4 EXPERIMENTS

We implemented the algorithm in section 3 and applied it to benchmark datasets to inspect the changes in accuracy and independence. Below, we present the details of the datasets and experimental conditions, and then provide experimental results.

4.1 Datasets

We used a Movielens 1M dataset (ML1M) [6] in our experiments. The number of users, items, and ratings were 6,040, 3,706, and 1,000,209, respectively. We regarded a user as preferring an item if the user gave the item a rating of 4 or higher.

We tested two types of sensitive features. The first, Year, represented whether a movie's release year was later than 1990. We selected this feature because it has been proven to influence preference patterns [15]. The sizes of ML1M-Year datasets whose sensitive values were 0 and 1 were 456,683 and 543,526, respectively. The second feature, Gender, represented the user's gender. The movie rating depended on the user's gender, and our recommender increased the independence of this information. The sizes of ML1M-Gender datasets whose sensitive values were 0 and 1 were 753,769 and 246,440, respectively. Comparing these two sensitive features, the sizes of ML1M-Year datasets divided by sensitive values were

more balanced than those of ML1M-Gender divided by sensitive values. The difference of original mean ratings between datasets, $\mathcal{D}^{(0)}$ and $\mathcal{D}^{(1)}$, is about five times larger in the ML1M-Year dataset than in the ML1M-Gender dataset.

4.2 Evaluation Indexes and Experimental Conditions

Next, we evaluated our experimental results in terms of prediction accuracy and the degree of independence. Prediction accuracy was measured by the area under the ROC curve (AUC) [4, 8]. This index measures how much more highly the relevant items are ranked in a recommendation list. A larger value of this index indicates better prediction accuracy.

We adopted two types of independence indexes. The first index measures the degree of independence between a sensitive feature and a preference score derived by equation (2). To evaluate the degree of independence, we checked the equality of the distributions of predicted ratings. For this purpose, we adopted the statistic of the two-sample Kolmogorov-Smirnov test (KS), which is a nonparametric test for the equality of two distributions. The KS statistic is defined as the area between two empirical cumulative distributions of predicted preferences for $\mathcal{D}^{(0)}$ and $\mathcal{D}^{(1)}$. A smaller KS indicates that R and S are more independent.

The second type of independence indexes is designed to evaluate the independence of a ranked list. We first assume that candidate items whose predicted preference scores are larger than a threshold are relevant items and the remaining items are irrelevant. A random variable, \tilde{R} , represents whether an item is relevant ($\tilde{R} = 1$) or irrelevant ($\tilde{R} = 0$), and \tilde{r} denotes its instance. The degree of independence between two binary variables, S and \tilde{R} , was evaluated by the following two indexes. Mutual information (MI) is defined as:

$$MI = \sum_{\tilde{r} \in \{0,1\}} \sum_{s \in \{0,1\}} \Pr[\tilde{r}, s] (\log \Pr[\tilde{r}, s] - \log \Pr[\tilde{r}] \Pr[s]), \quad (7)$$

and becomes 0 if \tilde{R} and S are perfectly independent. Calders & Verwer's discrimination score (CVS) [1] is defined as the probability of being relevant given $S=0$ subtracted by that given $S=1$,

$$CVS = \Pr[\tilde{R}=1|S=1] - \Pr[\tilde{R}=1|S=0], \quad (8)$$

and becomes 0 if \tilde{R} and S are perfectly independent.

The standard logistic PMF model and independence-enhanced logistic PMF model in section 3 were applied to the datasets in section 4.1. We tuned the hyper-parameters of the model so as to optimize the AUC obtained by a standard logistic PMF model. We used a regularization parameter, $\lambda = 0.1$, and dimension of cross terms, $K = 5$. We changed an independence parameter, η , from 10^{-2} to 10^2 and observed the accuracy and independence indexes. We performed a five-fold cross-validation procedure to obtain evaluation indexes for the accuracy and independence.

4.3 Experimental Results

In this experiment, we attempted to answer two questions. First, we examined whether or not our method as described in section 3 could actually enhance recommendation independence between a preference score and a sensitive feature. Second, in the case that independence of a preference score was enhanced, we analyzed whether the relevance of items was also independent.

To focus on the first question, whether our independence-enhancement method could enhance recommendation independence, we computed AUC and KS indexes by changing an independence parameter, η . Additionally, we showed the means of predicted preferences for two datasets, $\mathcal{D}^{(0)}$ and $\mathcal{D}^{(1)}$, in order to visualize how two the distributions were matched. Figures 2 and 3 show the experimental results. In terms of accuracy, Figures 2(a) and 3(a) show that the loss in accuracy measured by the AUC was very slight. These results were highly contrasted with those of our past experiments, in which the increase rate of error for the predicting-rating task was much higher. This may have been because, although the absolute values of predicted preference scores were changed, the relative rankings of scores among items were preserved. To examine this hypothesis, we compared pairs of predicted scores derived by our algorithms whose independence parameters were $\eta = 0.01$ and $\eta = 10$. The means of absolute differences were 0.053 (Year) and 0.025 (Gender), clearly indicating that the predicted scores were changed. Rank correlations (Spearman's ρ) between pairs of scores were extremely high, 0.978 (Year) and 0.990 (Gender). This observation means that the relative rankings among predicted scores were almost completely preserved, even if recommendation independence was enhanced, and thus the AUCs were not decreased because an AUC index was invariant for any monotonic transformations.

On the other hand, the independence between a predicted preference score and a sensitive feature was clearly enhanced in Figure 2(b). This claim could also be confirmed by the observation that the means of scores derived from $\mathcal{D}^{(0)}$ and $\mathcal{D}^{(1)}$ were made increasingly equal by increasing the parameter η in Figure 2(c). In Figure 3(b), it was unclear whether or not the index decreased, because the KS statistics were initially small. However, the matching of the two means in Figure 3(c) proved that the independence was enhanced. From the above, it may be concluded that recommendation independence of a preference score could be enhanced by our logistic PMF model, while the loss in accuracy was very slight.

We were thus able to confirm that the independence of a preference score, R , was enhanced. Next, we moved on to the second question, concerning the independence of the relevance of items from a sensitive feature. As described in section 4.2, we predicted preference scores for all user-item pairs in a dataset in a 5-fold cross-validation procedure, then ranked these items according as their scores are in descending order. In a find-good-items case, the top- k ranked items were assumed to be relevant, and were displayed to users. Hence, we have to take into account the enhancement of independence between a sensitive feature and an event whether a recommended item was relevant ($\tilde{R}=1$) or irrelevant ($\tilde{R}=0$). We then examine whether or not the enhancement between R and S could enhance the independence between \tilde{R} and S . To examine the independence, we computed the independence indexes as shown in equations (7) and (8) at various threshold of k . Figures 4 and 5 show the changes in the independence indexes according to the number of relevant items, k . By enhancing the independence of preference scores, the independence in regard to relevance was also enhanced for most of the values of k , when compared with a standard recommender. However, the independence of relevance

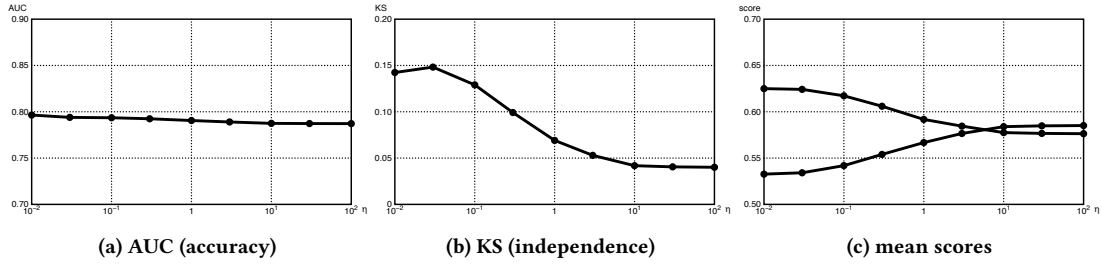


Figure 2: Changes of accuracy and independence indexes for the ML1M-Year dataset

NOTE : These figures show the changes of indexes according to an independence parameter, η . The X-axes represent the independence parameter in a logarithmic scale. The Y-axis of the subfigure (a) shows an AUC index to evaluate prediction accuracy. The Y-axis of the subfigure (b) shows the Kolmogorov-Smirnov (KS) statistic to evaluate recommendation independence. Larger AUC indicates better performance in accuracy, and smaller KS indicates better performance in independence. Subfigure (c) shows the means of predicted preference scores for the datasets, $\mathcal{D}^{(0)}$ and $\mathcal{D}^{(1)}$.

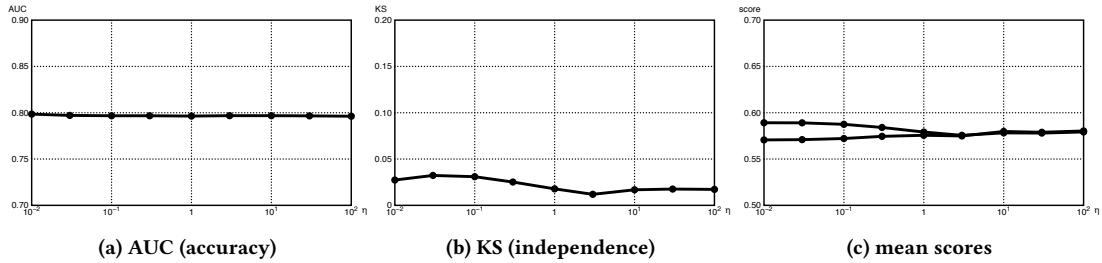


Figure 3: Changes of accuracy and independence indexes for the ML1M-Gender dataset

NOTE : See the note for Figure 2.

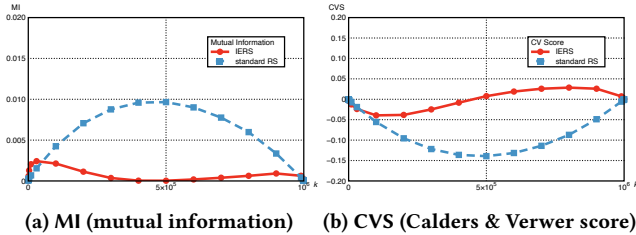


Figure 4: Changes of independence between \tilde{R} and S for the ML1M-Year dataset

NOTE : These figures show the changes of independence indexes according to the number of relevant items. The X-axes represent the number of relevant items, k . The Y-axis of the subfigure (a) shows mutual information (equation (7)). Blue broken lines show the changes of independence obtained by a standard recommendation algorithm, and red solid lines show the changes obtained by our independence-enhanced recommendation algorithm. A relevance variable, \tilde{R} , and a sensitive feature, S , are completely independent if the mutual information is zero. The Y-axis of the subfigure (b) shows Calders and Verwer’s discrimination indexes (equation (8)). These indexes are exactly zero if \tilde{R} and S are independent.

was not enhanced for small k in both datasets and indexes. Unfortunately, because users cannot check many items, independence for small k is very important. Therefore, this failure to enhance independence was a serious issue. From this experiment, the enhancement of independence in regard to preference scores did not always enhance independence of relevance.

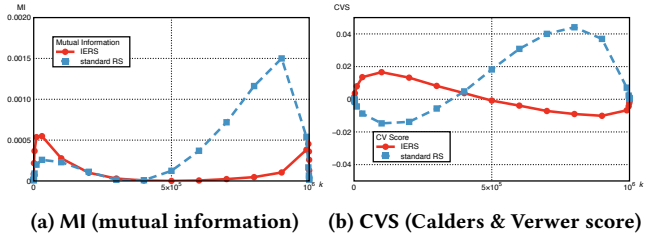


Figure 5: Changes of independence between \tilde{R} and S for the ML1M-Gender dataset

NOTE : The note for Figure 4 applies, except that the scaling of Y-axes is changed to clarify the differences of independence indexes.

The experimental results could be summarized as follows:

- Our algorithm could successfully enhance independence between a preference score and a sensitive feature, without appreciably decreasing the accuracy compared to a predicting-ratings case.
- The independence in terms of relevance might not always be enhanced by enhancing the independence of a preference score.

From these experimental results, we conclude that a method must be specially designed to enhance independence between item relevance and sensitive information.

5 RELATED WORK

We wish to emphasize that recommendation independence is distinct from recommendation diversity [16, 23]. First, while diversity may be the property of a set of recommendations, independence is a relation between each recommendation and a sensitive feature. Second, recommendation independence depends on the specification of a sensitive feature, while recommendation diversity depends on the specification of a similarity metric between a pair of items. Finally, while diversity seeks to provide a wider range of topics, independence seeks to provide unbiased information.

We adopted techniques for fairness-aware data mining to enhance the independence. Fairness-aware data mining is a general term for mining techniques designed so that sensitive information does not influence the mining results. Pedreschi et al. first advocated such mining techniques, which emphasized the unfairness in association rules whose consequents include serious determinations [17]. Another technique of fairness-aware data mining focuses on predictions designed so that the influence of sensitive information on the predictions is reduced [1, 10]. These techniques would be directly useful in the development of an independence-enhanced variant of content-based recommender systems, because content-based recommenders can be implemented by standard classifiers. Specifically, class labels indicate whether or not a user prefers an item, and the features of objects correspond to features of the item.

The concept behind recommendation transparency is that it might be advantageous to explain the reasoning underlying individual recommendations. Indeed, such transparency has been proven to improve the satisfaction of users [20], and different methods of explanation have been investigated [7]. In the case of recommendation transparency, the system tries to persuade users of its objectivity by demonstrating that the recommendations were not made by any malicious manipulations. On the other hand, in the case of independence, the objectivity is guaranteed by satisfying a previously defined regulation, i.e., recommendation independence.

6 CONCLUSIONS

We previously developed a method to enhance recommendation independence for a predicting-ratings task. In this paper, we examined recommendation independence for a find-good-items task. We designed a new model to enhance independence of a predicted preference score from a sensitive feature. We empirically showed that this model could enhance independence from a preference score, but the losses in accuracy were very slight. We further examined independence in terms of the relevance of recommended items, but this type of independence sometimes failed to be enhanced.

There are many functionalities required for an IERS. From our experimental results, we must consider a new notion of recommendation independence in terms of a ranked recommendation list for a find-good-items task. Because in this paper we assumed that users explicitly rate the relevance of items, we have to develop a method applicable to the case of implicit ratings. However, it would be difficult to select which items should be treated as irrelevant, because such selection would influence the state of independence. Bayesian extension would not be straightforward because the parameters are probabilistically generated and recommendation independence might be violated under specific choices of parameters. Because

sensitive features are currently restricted to binary types, we will try to deal with sensitive features whose types are multivariate discrete or continuous.

7 ACKNOWLEDGMENTS

We gratefully acknowledge the valuable comments and suggestions of Dr. Yoshinori Hijikata. We would also like to thank the GroupLens research lab for providing datasets. This work is supported by MEXT/JSPS KAKENHI Grant Number JP24500194, JP15K00327, and JP16H02864.

REFERENCES

- [1] T. Calders and S. Verwer. 2010. Three naive Bayes Approaches for Discrimination-free Classification. *Data Mining and Knowledge Discovery* 21 (2010), 277–292.
- [2] Ö. Celma and P. Cano. 2008. From Hits to Niches?: or How Popular Artists Can Bias Music Recommendation and Discovery. In *Proc. of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*.
- [3] S. Forden. 2012. Google Said to Face Ultimatum From FTC in Antitrust Talks. Bloomberg. (Nov. 13 2012). (<http://bloom.bg/PPNEaS>).
- [4] A. Gunawardana and G. Shani. 2009. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research* 10 (2009), 2935–2962.
- [5] S. Hajian, F. Bonchi, and C. Castillo. 2016. Algorithmic Bias: from Discrimination Discovery to Fairness-Aware Data Mining. The 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, Tutorial. (2016).
- [6] F. M. Harper and J. A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. on Interactive Intelligent Systems* 5, 4 (2015).
- [7] J. L. Herlocker, J. A. Konstan, and J. Riedl. 2000. Explaining Collaborative Filtering Recommendations. In *Proc. of the Conf. on Computer Supported Cooperative Work*. 241–250.
- [8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. on Information Systems* 22, 1 (2004), 5–53.
- [9] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. 2012. Enhancement of the Neutrality in Recommendation. In *The 2nd Workshop on Human Decision Making in Recommender Systems*.
- [10] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. 2012. Fairness-aware Classifier with Prejudice Remover Regularizer. In *Proc. of the ECML PKDD 2012, Part II*. 35–50. [LNCS 7524].
- [11] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. 2013. Efficiency Improvement of Neutrality-enhanced Recommendation. In *The 3rd Workshop on Human Decision Making in Recommender Systems*.
- [12] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. 2014. Correcting Popularity Bias by Enhancing Recommendation Neutrality. In *The 8th ACM Conference on Recommender Systems, Poster*.
- [13] T. Kamishima, S. Akaho, H. Asoh, and I. Sato. 2016. Model-Based Approaches for Independence-Enhanced Recommendation. In *Proc. of the IEEE 16th Int'l Conf. on Data Mining Workshops*. 860–867.
- [14] Y. Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. 426–434.
- [15] Y. Koren. 2009. Collaborative Filtering with Temporal Dynamics. In *Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. 447–455.
- [16] S. M. McNee, J. Riedl, and J. A. Konstan. 2006. Accurate Is Not Always Good: How Accuracy Metrics Have Hurt Recommender Systems. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. 1097–1101.
- [17] D. Pedreschi, S. Ruggieri, and F. Turini. 2008. Discrimination-aware Data Mining. In *Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. 560–568.
- [18] P. Resnick, J. Konstan, and A. Jameson. 2011. Panel on The Filter Bubble. The 5th ACM Conf. on Recommender Systems. (2011). (<http://acmrecsys.wordpress.com/2011/10/25/panel-on-the-filter-bubble/>).
- [19] R. Salakhutdinov and A. Mnih. 2008. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems* 20. 1257–1264.
- [20] R. Sinha and K. Swearingen. 2002. The Role of Transparency in Recommender Systems. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. 830–831.
- [21] L. Sweeney. 2013. Discrimination in Online Ad Delivery. *Commun. ACM* 56, 5 (2013), 44–54.
- [22] S. Watanabe. 1969. *Knowing and Guessing – Quantitative Study of Inference and Information*. John Wiley & Sons.
- [23] C. N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. 2005. Improving Recommendation Lists Through Topic Diversification. In *Proc. of the 14th Int'l Conf. on World Wide Web*. 22–32.